

**UNISYS**

**B 2000/B 3000/  
B 4000/V Series  
DMSII**

**Operations  
Reference Manual**

**Volume 2: Database  
Administration**

Copyright © 1987 Unisys Corporation  
All Rights Reserved  
Unisys is a trademark of Unisys Corporation

Relative to Release  
Level 7.2

Priced Item

December 1987  
Distribution Code SD  
Printed in U S America  
5023039



The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual living or otherwise, or that of any group or association is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Comments or suggestions regarding this document should be submitted on a Field Communication Form (FCF) with the CLASS specified as 2 (S.W.:System Software), and the Type specified as 3 (DOC), and the product specified as the 7-digit form number of the manual (for example, 5023039).





## TABLE OF CONTENTS

| Section | Title                                                     | Page |
|---------|-----------------------------------------------------------|------|
|         | ABOUT THIS MANUAL .....                                   | xi   |
|         | PURPOSE, SCOPE, AND AUDIENCE .....                        | xi   |
|         | ORGANIZATION .....                                        | xi   |
|         | RELATED DOCUMENTS .....                                   | xii  |
| 1       | DATABASE FILES OVERVIEW .....                             | 1-1  |
|         | INTRODUCTION .....                                        | 1-1  |
|         | FILE NAMING CONVENTIONS .....                             | 1-1  |
|         | THE DATABASE PROGRAM .....                                | 1-1  |
|         | DATABASE PROGRAM TAILORED CODE ICM .....                  | 1-2  |
|         | DATA DEFINITION FILE .....                                | 1-2  |
|         | CONTROL FILE .....                                        | 1-2  |
|         | STATISTICS FILE .....                                     | 1-2  |
|         | STRUCTURE FILES .....                                     | 1-2  |
|         | AUDIT TRAIL FILES .....                                   | 1-3  |
|         | MEMORY FILE .....                                         | 1-3  |
| 2       | ADMINISTRATIVE CONTROL INTERFACES .....                   | 2-1  |
|         | INTRODUCTION .....                                        | 2-1  |
|         | DMS2 Option .....                                         | 2-1  |
|         | LIMIT Records .....                                       | 2-1  |
|         | System Commands .....                                     | 2-2  |
|         | IR Command .....                                          | 2-3  |
|         | HALTLOAD Option .....                                     | 2-7  |
|         | REBUILD Option .....                                      | 2-7  |
|         | ROLLBACK Option .....                                     | 2-8  |
|         | RECONSTRUCT Option .....                                  | 2-10 |
|         | REORGANIZE Option .....                                   | 2-10 |
|         | PRODUCTION Option .....                                   | 2-12 |
|         | PURGE Option .....                                        | 2-13 |
|         | LOAD Option .....                                         | 2-13 |
|         | DUMP Option .....                                         | 2-13 |
|         | RESTART Option .....                                      | 2-14 |
|         | SB Command .....                                          | 2-15 |
|         | SP Command .....                                          | 2-16 |
| 3       | UPDATE, REORGANIZATION, AND PURGING .....                 | 3-1  |
|         | INTRODUCTION .....                                        | 3-1  |
|         | UPDATE .....                                              | 3-1  |
|         | Performing Update .....                                   | 3-1  |
|         | REORGANIZATION .....                                      | 3-7  |
|         | Precautions and Limitations .....                         | 3-10 |
|         | Database Maintenance .....                                | 3-11 |
|         | Performing Database Maintenance with DASDL (XXXREO) ..... | 3-11 |
|         | Performing Database Maintenance with DMSREO .....         | 3-12 |
|         | File Format Conversion .....                              | 3-13 |
|         | Record Format Conversion .....                            | 3-17 |
|         | Version Checking .....                                    | 3-20 |
|         | Record Format Change Pragmatics .....                     | 3-21 |
|         | Data Item Changes .....                                   | 3-21 |



## TABLE OF CONTENTS (Cont.)

| Section | Title                                                | Page |
|---------|------------------------------------------------------|------|
|         | ADD/DELETE Items .....                               | 3-21 |
|         | Item Options .....                                   | 3-22 |
|         | Structure Changes .....                              | 3-22 |
|         | Performing Record Format Conversion .....            | 3-23 |
|         | RESTART .....                                        | 3-27 |
|         | PURGING .....                                        | 3-29 |
|         | RELEASE UPDATES .....                                | 3-29 |
|         | CHANGES DASDL PERMITS .....                          | 3-31 |
| 4       | AUDIT AND RECOVERY .....                             | 4-1  |
|         | INTRODUCTION .....                                   | 4-1  |
|         | TRANSACTIONS AND SYNCPOINTS .....                    | 4-1  |
|         | CONTROLPOINTS AND THE FLUSHING OF DATA BUFFERS ..... | 4-2  |
|         | AUDIT BUFFER AND DATA BUFFER SYNCHRONIZATION .....   | 4-3  |
|         | AUDIT CHANGE NUMBERS AND AUDIT BLOCK NUMBERS .....   | 4-3  |
|         | TYPES OF RECOVERY .....                              | 4-4  |
|         | Program Abort Recovery .....                         | 4-4  |
|         | Halt/Load Recovery .....                             | 4-5  |
|         | Rebuild Recovery and Reconstruct Recovery .....      | 4-6  |
|         | Rebuild Recovery .....                               | 4-6  |
|         | Reconstruct Recovery .....                           | 4-7  |
|         | Offline Reconstruct .....                            | 4-7  |
|         | Online Reconstruct .....                             | 4-7  |
|         | Details of Operation .....                           | 4-7  |
|         | Rollback Recovery .....                              | 4-8  |
|         | RESTART DATA SETS .....                              | 4-8  |
| 5       | HOW TO DUMP AND RELOAD DATABASES .....               | 5-1  |
|         | INTRODUCTION .....                                   | 5-1  |
|         | FILES .....                                          | 5-1  |
|         | DUMPING TO TAPE .....                                | 5-3  |
|         | Full Database Dump .....                             | 5-3  |
|         | Partial Database Dump .....                          | 5-3  |
|         | Offline Dump .....                                   | 5-4  |
|         | Online Dump .....                                    | 5-4  |
|         | RECOVERY USING DATABASE BACKUPS .....                | 5-4  |
|         | LOAD Option .....                                    | 5-4  |
|         | REBUILD Option .....                                 | 5-5  |
|         | RECONSTRUCT Option .....                             | 5-5  |
|         | BACKING UP WITH THE SYSTEM/COPY UTILITY .....        | 5-5  |
|         | PROCEDURES .....                                     | 5-6  |
|         | Using the Menu Screens (DMSUTL) .....                | 5-6  |
|         | Using the IR Command .....                           | 5-7  |
| 6       | PRODUCTION DATABASE PROGRAMS .....                   | 6-1  |
|         | INTRODUCTION .....                                   | 6-1  |
|         | GENERATING A PRODUCTION DBP .....                    | 6-1  |
|         | INITIATING THE PRODUCTION DBP .....                  | 6-4  |
|         | DETAILS .....                                        | 6-5  |



### TABLE OF CONTENTS (Cont.)

| Section | Title                                                  | Page |
|---------|--------------------------------------------------------|------|
| 7       | SYSTEM MESSAGES AND ERROR MESSAGES FOR OPERATORS ..... | 7-1  |
|         | INTRODUCTION .....                                     | 7-1  |
|         | MESSAGES .....                                         | 7-1  |
| 8       | EXCEPTION CONDITIONS FOR USER PROGRAMS .....           | 8-1  |
|         | INTRODUCTION .....                                     | 8-1  |
|         | NOTES .....                                            | 8-13 |
| 9       | DETAILS OF DATABASE FILES .....                        | 9-1  |
|         | INTRODUCTION .....                                     | 9-1  |
|         | CONTROL FILE FORMAT .....                              | 9-1  |
|         | Header Record .....                                    | 9-1  |
|         | Structure Records .....                                | 9-4  |
|         | STATISTICS FILE FORMAT .....                           | 9-9  |
|         | Header Record .....                                    | 9-9  |
|         | Database Record .....                                  | 9-9  |
|         | Structure Records .....                                | 9-11 |
|         | User Records .....                                     | 9-13 |
|         | MEMORY FILE FORMAT .....                               | 9-14 |
|         | AUDIT TRAIL FILE FORMAT .....                          | 9-15 |
|         | Header Record .....                                    | 9-16 |
|         | Trailer Record .....                                   | 9-17 |
|         | Event Records .....                                    | 9-18 |
| 10      | DATABASE PROGRAM (DBP) INTERNAL STRUCTURES .....       | 10-1 |
|         | INTRODUCTION .....                                     | 10-1 |
|         | DBP MEMORY REQUIREMENT .....                           | 10-1 |
|         | MEMORY SIZE ESTIMATE .....                             | 10-4 |
|         | OTHER CONSIDERATIONS .....                             | 10-5 |
| 11      | USER PROGRAM INTERFACE TO DMSII .....                  | 11-1 |
|         | INTRODUCTION .....                                     | 11-1 |
|         | STRUCTURE INFORMATION BLOCK (SIB) .....                | 11-1 |
|         | DMSTATUS REGISTER .....                                | 11-3 |
|         | EXCEPTION HANDLING METHODS .....                       | 11-3 |
|         | SET SELECTION EXPRESSION STRING .....                  | 11-4 |
|         | Code Structure .....                                   | 11-4 |
|         | Logical Container .....                                | 11-4 |
|         | Relational Container .....                             | 11-4 |
|         | Relational OP [1 digit] .....                          | 11-5 |
|         | Key Item Identifiers [2 digits] .....                  | 11-5 |
|         | Value Offset [4 digits] .....                          | 11-5 |
|         | Value Type [1 digit] .....                             | 11-5 |
|         | Value Length [3 digits] .....                          | 11-5 |
|         | Value Pool Structure .....                             | 11-6 |
|         | INDEX .....                                            | 1    |



## LIST OF ILLUSTRATIONS

| Figure | Title                                                                      | Page |
|--------|----------------------------------------------------------------------------|------|
| 2-1    | IR Command Syntax .....                                                    | 2-3  |
| 2-2    | IR Command Parameters Syntax .....                                         | 2-3  |
| 2-3    | IR Command Options Syntax .....                                            | 2-4  |
| 2-4    | Database Activity (SB) Command Syntax .....                                | 2-15 |
| 2-5    | SP Command Syntax .....                                                    | 2-16 |
| 2-6    | SP Command Parameters .....                                                | 2-17 |
| 3-1    | Update Compile .....                                                       | 3-2  |
| 3-2    | Reorganization Compile .....                                               | 3-8  |
| 3-3    | Reorganization Phase .....                                                 | 3-9  |
| 3-4    | Example of DASDL Source for a Database Maintenance<br>Reorganization ..... | 3-11 |
| 3-5    | Sample Databases During Record Format Conversion .....                     | 3-19 |
| 6-1    | DMSICM Syntax .....                                                        | 6-1  |
| 6-2    | REBIND PRODUCTION Syntax .....                                             | 6-2  |
| 6-3    | Production DBP Creation .....                                              | 6-3  |
| 6-4    | Production DBP Execution .....                                             | 6-4  |





## LIST OF TABLES

| Table | Title                                                          | Page |
|-------|----------------------------------------------------------------|------|
| 3-1   | Record Format Conversion .....                                 | 3-18 |
| 3-2   | Mappings of PERSON Between Databases in Figure 3-5 .....       | 3-20 |
| 3-3   | Converting Databases for New Software Release .....            | 3-30 |
| 3-4   | Allowed Changes to Options .....                               | 3-31 |
| 3-5   | Allowed Changes to Parameters .....                            | 3-31 |
| 3-6   | Allowed Changes to Audit Trail .....                           | 3-31 |
| 3-7   | Allowed Changes to Data Sets .....                             | 3-33 |
| 3-8   | Allowed Changes to Sets, Subsets, and Accesses .....           | 3-36 |
| 3-9   | Allowed Changes to Remaps .....                                | 3-39 |
| 5-1   | Dumping and Loading Database Files .....                       | 5-2  |
| 8-1   | Exception Categories .....                                     | 8-1  |
| 9-1   | Control File Header Record (Block One) .....                   | 9-1  |
| 9-2   | Control File Structure Records (Blocks 2-n) .....              | 9-5  |
| 9-3   | Control File Family Attributes Field .....                     | 9-8  |
| 9-4   | Statistics File Database Record .....                          | 9-9  |
| 9-5   | Statistics File Structure Records (Blocks 3-n) .....           | 9-12 |
| 9-6   | Statistics File User Record Format .....                       | 9-13 |
| 9-7   | Audit File Header Record (Block One) .....                     | 9-16 |
| 9-8   | Audit File Trailer Record Format .....                         | 9-17 |
| 9-9   | Audit File Syncpoint Record .....                              | 9-18 |
| 9-10  | Audit File Controlpoint Record .....                           | 9-18 |
| 9-11  | Audit File Database OPEN Record .....                          | 9-19 |
| 9-12  | Audit File Database CLOSE Record .....                         | 9-19 |
| 9-13  | Audit File Database Program BOJ Record .....                   | 9-20 |
| 9-14  | Audit File Database Program EOJ Record .....                   | 9-20 |
| 9-15  | Audit File Change Record .....                                 | 9-20 |
| 9-16  | Audit File Recovery Record .....                               | 9-21 |
| 9-17  | Audit File Discontinuity Record .....                          | 9-21 |
| 9-18  | Audit File STORE (New) Record .....                            | 9-22 |
| 9-19  | Audit File DELETE (Old) Record .....                           | 9-22 |
| 9-20  | Audit File Data Set Update Record .....                        | 9-23 |
| 9-21  | Audit File Before Image of Data Set BCI Record .....           | 9-23 |
| 9-22  | Audit File After Image of Data Set BCI Record .....            | 9-24 |
| 9-23  | Audit File Random Data Set STORE (New) Record .....            | 9-24 |
| 9-24  | Audit File Random Data Set DELETE (Old) Record .....           | 9-25 |
| 9-25  | Audit File Record of Entry Added to Table .....                | 9-25 |
| 9-26  | Audit File Record Deleting Entry from Table .....              | 9-26 |
| 9-27  | Audit File Record of Updating Table Entry .....                | 9-26 |
| 9-28  | Audit File Record for Table Split .....                        | 9-27 |
| 9-29  | Audit File Record for Root Table Split .....                   | 9-27 |
| 9-30  | Audit File Record of Before Image of Table BCI .....           | 9-28 |
| 9-31  | Audit File Record of After Image of Table BCI .....            | 9-28 |
| 9-32  | Audit File Record of Block One Update .....                    | 9-29 |
| 9-33  | Audit File Record of General Before Image .....                | 9-29 |
| 9-34  | Audit File Record of General After Image .....                 | 9-30 |
| 9-35  | Audit File Record of Data Write Error .....                    | 9-30 |
| 9-36  | Audit File Record of Data Read Error .....                     | 9-31 |
| 9-37  | Audit File Record of Initialize to Random Data Set Block ..... | 9-31 |
| 9-38  | Audit File Record of PURGE Structure List .....                | 9-32 |
| 9-39  | Audit File Record of PURGE Done .....                          | 9-32 |



**LIST OF TABLES (Cont.)**

| <b>Table</b> | <b>Title</b>                                        | <b>Page</b> |
|--------------|-----------------------------------------------------|-------------|
| 9-40         | Audit File Record of Start Online Dump .....        | 9-32        |
| 9-41         | Audit File Record of End Online Dump .....          | 9-33        |
| 9-42         | Audit File Record of BEGIN-TRANSACTION .....        | 9-34        |
| 9-43         | Audit File Record of END-TRANSACTION .....          | 9-34        |
| 10-1         | Database Program Structures .....                   | 10-3        |
| 11-1         | Logical Database Identification Record Format ..... | 11-2        |
| 11-2         | Structure Identification Record Format .....        | 11-2        |



## ABOUT THIS MANUAL

### PURPOSE, SCOPE, AND AUDIENCE

The Data Management System II (DMSII) is a software product that enables a Database Administrator (DBA) to create and maintain a database. This manual describes the tasks the Database Administrator and operators use to maintain databases. In addition, this manual covers exception conditions for user programs and error messages and system messages for operators.

This manual is written for Database Administrators and DMSII operators. This manual is part of a three volume set that includes the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation* and the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*. To use this manual, you need to be familiar with general database management concepts and the Data Management System II (DMSII).

### ORGANIZATION

This document consists of the following 11 sections:

#### SECTION 1: DATABASE FILES OVERVIEW

This section gives an overview of the database files that DASDL creates and the naming conventions DASDL uses.

#### SECTION 2: ADMINISTRATIVE CONTROL INTERFACES

This section describes the interfaces between the Master Control Program (MCP) and the DMSII software.

#### SECTION 3: UPDATE AND REORGANIZATION

This section describes how to change the database with the update and reorganization processes.

#### SECTION 4: AUDIT AND RECOVERY

This section describes the audit and recovery processes.

#### SECTION 5: HOW TO DUMP AND RELOAD DATABASES

This section tells how to dump and reload a database for the recovery process.

#### SECTION 6: PRODUCTION DATABASE PROGRAMS

This section describes Production database programs (DBPs). A Production DBP contains object code of both a DBP and a user program.

#### SECTION 7: SYSTEM MESSAGES AND ERROR MESSAGES FOR OPERATORS

This section includes the system messages and error messages that do not apply to the utilities. All messages that apply to the utilities appear in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*.

#### SECTION 8: EXCEPTION CONDITIONS FOR USER PROGRAMS

This section includes the category number, subcategory name, number, and description of all of the exception conditions that can occur during the execution of a DBP.



## SECTION 9: DETAILS OF DATABASE FILES

This section describes the file formats for each of the database files.

## SECTION 10: DATABASE PROGRAM (DBP) INTERNAL STRUCTURES

This section describes the internal structures of the DBP. The DBP performs all of the functions a user program needs to access and update a particular database.

## SECTION 11: USER PROGRAM INTERFACE TO DMSII

This section tells about the user program interface with the DMSII software.

## RELATED DOCUMENTS

The following related publications are listed, for reference, according to the operating system to which they apply.

*B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation* for MCPIX, MCP/VS 1.0, and MCP/VS 2.0.

*B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities* for MCPIX, MCP/VS 1.0, and MCP/VS 2.0.

*B 2000/B 3000/B 4000/V Series DMSII Inquiry Operations Guide* and the *DMINQUIRY Reference Card* for MCPIX and MCP/VS 1.0; or the *B 2000/B 3000/B 4000/V Series DMSII Inquiry Operations Guide* for MCP/VS 2.0.

*2000/B 3000/B 4000 Series MCPIX System Software Operations Guide Volume 1: System Commands* and *B 2000/B 3000/B 4000 Series MCPIX System Software Operations Guide Volume 2: System Utilities* for MCPIX or *V Series MCP/VS System Software Operations Guide Volume 2: System Commands* and *V Series MCP/VS System Software Operations Guide Volume 3: System Utilities* for MCP/VS 2.0.

*B 2000/B 3000/B 4000 Series MCPIX System Software Operations Guide Volume 1: System Commands* for MCPIX or *V Series MCP/VS System Software Operations Guide Volume 1: Installation* for MCP/VS 2.0.

*B 2000/B 3000/B 4000/V Series CANDE Installation and Operation Guide* for MCPIX and MCP/VS 1.0; or the *V Series CANDE Installation and Operations Guide* for MCP/VS 2.0.

*B 2000/B 3000/B 4000 Series COBOL ANSI-74 Compiler Programming Reference Manual* for MCPIX and MCP/VS 1.0; or the *V Series COBOL ANSI-74 Compiler Programming Reference Manual* for MCP/VS 2.0.

*B 2000/B 3000/B 4000/V Series Pascal Compiler Programming Reference Manual* for MCPIX, MCP/VS 1.0, and MCP/VS 2.0.

*B 2000/B 3000/B 4000/V Series RPG Compiler Programming Reference Manual* for MCPIX, MCP/VS 1.0, and MCP/VS 2.0.

*B 2000/B 3000/B 4000/V Series Programmer's Guide* for MCPIX and MCP/VS 1.0; or the *V Series MCP/VS Programming Reference Manual* for MCP/VS 2.0.





## SECTION 1

# DATABASE FILES OVERVIEW

## INTRODUCTION

This section provides an overview of database files.

## FILE NAMING CONVENTIONS

DASDL creates the following files:

- XXXXXX = Database program (DBP) code file
- XXXICM = Tailored code ICM
- XXXDDF = Data Definition File (DDF)
- XXXCNF = Control file
- XXX001 = Statistics file
- XXX003-XXX999 = Structure files
- XXXA01-XXXM99 = Audit trail files
- XXXN01-XXXZ99 = Duplicate audit trail files
- XXXMEM = Memory file

The Data Management System II (DMSII) uses the database name for the names of many of these files. The DASDL compiler establishes the database name in the COMPILE statement when it compiles the DASDL source.

## THE DATABASE PROGRAM

The name of the database program (DBP) is the name of the database that you specified in the COMPILE statement. This name must be at least three characters because the system uses the first three characters to name other database files. The DBP resides on disk and must be present when a user program opens the database.

If you update a database description in such a manner that the reorganization process is required, the DASDL Compiler creates an alternate database program, which resides on disk, and is named XXXREO where XXX represents the first three letters of the database name. Refer to Section 3 for more information about the reorganization process.



## DATABASE PROGRAM TAILORED CODE ICM

The Database Program Tailored Code ICM (Independently Compiled Module) contains code that is specifically tailored to the characteristics of this database. It is used in the DASDL binding phase along with generalized database ICMs to create the DBP. The resulting file, which resides on disk, is named XXXICM where XXX represents the first three letters of the database name. The DASDL Compiler needs this file on disk for a REBIND of the DBP.

If you modify a database description in such a manner that the reorganization process is required, the DASDL Compiler creates a second database program tailored code file. This second tailored code file, which resides on disk, is named XXXRCM where XXX represents the first three letters of the database name. Refer to Section 3 for more information about the reorganization process.

## DATA DEFINITION FILE

The DBP uses the Data Definition File (DDF) to hold information about the database structures and the items within those structures. The DDF, which resides on disk or diskpack, is named XXXDDF where XXX represents the first three characters of the database name.

## CONTROL FILE

The DBP uses the control file as a depository for various dynamic values that it needs from run to run. Whenever a DBP is activated, a creation date-timestamp in the DBP is also checked against a similar date-timestamp in the control file and each invoked structure file for compatibility. The control file is named XXXCNF where XXX represents the first three characters of the database name. The file XXXCNF resides on the disk or diskpack specified by the FAMILYNAME statement of the database identifier. For more information about FAMILYNAME refer to <physical attributes description> in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*.

## STATISTICS FILE

The statistics file contains various run-time statistics for the database. The statistics file resides on disk or diskpack and is structure number 1. The statistics file is named XXX001 where XXX represents the first three characters of the database name.

## STRUCTURE FILES

Each database structure (data set, set, and subset) resides in a separate file on disk or diskpack. The DASDL compiler assigns each structure a structure number between 003 and 999. The file name for each structure is then XXXNNN where XXX represents the first three letters of the database name and NNN is the structure number (003-999).



## AUDIT TRAIL FILES

The DBP uses the audit trail files to maintain before images and after images of an audited database. The audit trail resides on disk or diskpack with a file name of XXXYNN, where:

- XXX = the first three letters of the database name
- Y = a letter, A through M, for primary audit and, N through Z, for  
secondary (DUPLICATED) audit trail
- NN = a 2-digit number from 01 to 99

Refer to Section 4 for more information on audit trail files.

## MEMORY FILE

The DASDL Compiler uses the memory file to store various tables and parameters to be input to the DBP at Beginning Of Job (BOJ). The memory file resides on the same medium as the control file. Its file name is XXXMEM, where XXX represents the first three letters of the database name.



## SECTION 2

### ADMINISTRATIVE CONTROL INTERFACES

#### INTRODUCTION

To prepare the Master Control Program (MCP) for the use of the Data Management System II (DMSII), you must attend to the following items:

- The DMS2 option
- LIMIT records in the system configuration file
- The IR, SB, and SP commands

#### DMS2 Option

You must set the DMS2 option in the MCP before you can use DMSII. There are two ways to do this:

- Include a USE DMS2 record in the system configuration file. This takes effect the next time you do a halt/load or a cold start. If you include a USE DMS2 record, you must specify LIMIT DMSDBP and LIMIT DMSUSERS in the system configuration file.
- Enter the SO DMS2 command at the ODT. This takes effect immediately.

#### LIMIT Records

Four LIMIT records in the system configuration file affect DMSII:

LIMIT DMSDBP <integer>

Use the LIMIT DMSDBP record to specify the maximum number of database programs allowed to run at one time. This is equivalent to the number of databases allowed open at one time. The minimum value of integer is 1. When you are running under MCPIX, ASR 7.0, the maximum value is 9. When you are running under MCP/VS, mark 1.0 or 2.0, the maximum value is 98. If you include the DMSUSERS record in the system configuration file, but do not specify DMSDBP, the default value for DMSDBP limit is 1.





#### LIMIT DMSUSERS <integer>

Use the LIMIT DMSUSERS record with the LIMIT DMSDBP record to control the maximum number of DMS programs allowed to run in the mix at any one time. These LIMIT records control the size of specific internal DMS tables. You must include this record, if you set the DMS2 option. The default value is 0.

#### NOTE

The sum of the DMSDBP limit and the DMSUSERS limit cannot exceed 98.

#### LIMIT DMSTATUS <integer>

Use the LIMIT DMSTATUS record to specify the STATUS interval for the database program (DBP). The integer must be from 1 to 90 inclusive. Each incremental value represents 10 seconds of actual time (that is, 1 = 10 seconds, 90 = 900 seconds or 15 minutes, etc.). The default value is 30 (5 minutes). The DBP performs garbage collection (for example, to return unused memory and buffers) on every other STATUS poll.

#### LIMIT PRIORITYQ <integer>

Use the LIMIT PRIORITYQ record to specify the "center point" for queuing of requests for operating system services. This center point affects requests from all programs including DMSII user programs. The operating system queues all requests from programs with a priority above the center point in priority order. Requests with priorities at or below the center point will all receive the same priority.

### System Commands

Use the following system commands to obtain information about the database, to set certain database parameters, and to initiate certain DMSII functions:

- IR = Initiate recovery, reorganization, dump/load, purge, or production DBP.
- SB = Display database activity.
- SP = Display/set/change database parameters.

The full syntax and semantics of the IR, SB, and SP commands appear later in this section.



## IR Command

The IR command initiates the following processes: recovery, reorganization of database files, execution of production DBPs, purging of database structures, and loading and dumping of archive tapes.

### NOTE

All of the functions of the IR command are available through the screen-driven utility interface described in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*.

The syntax for the IR command appears in Figure 2-1. The syntax for the parameters and options that you use with the IR command appear in Figures 2-2 and 2-3.

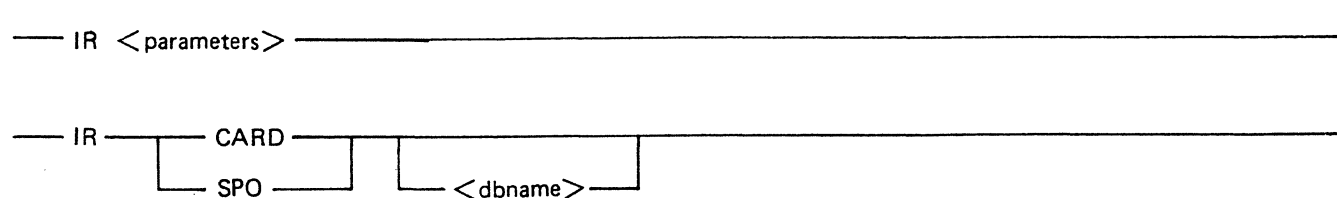


Figure 2-1. IR Command Syntax

When you specify CARD or SPO, you enter the parameters by means of a card file or with AX (ACCEPT) commands, respectively. If you enter the optional database name, <dbname>, the database name will appear as the multifile ID when DMSDLR is initiated (that is, <dbname/DMSDLR>). If you use CARD, the DMSDLR program will read a card file labelled DLRIN. If you specify SPO, DMSDLR will request the data by means of AX commands, and you must enter END to terminate the AX command input.

The maximum length of text for the IR command is 100 characters. If the desired command is longer than this, you must use the CARD or SPO option.

The IR command invokes the DMSDLR program, which performs recovery, reorganization, purging, backup, loading, and restart functions.

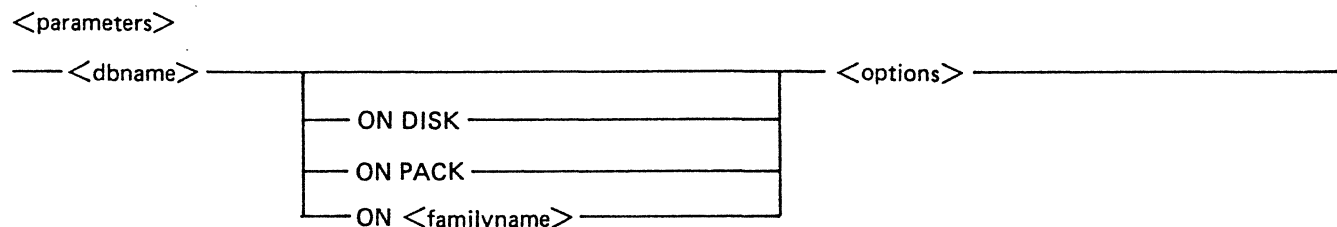


Figure 2-2. IR Command Parameters Syntax

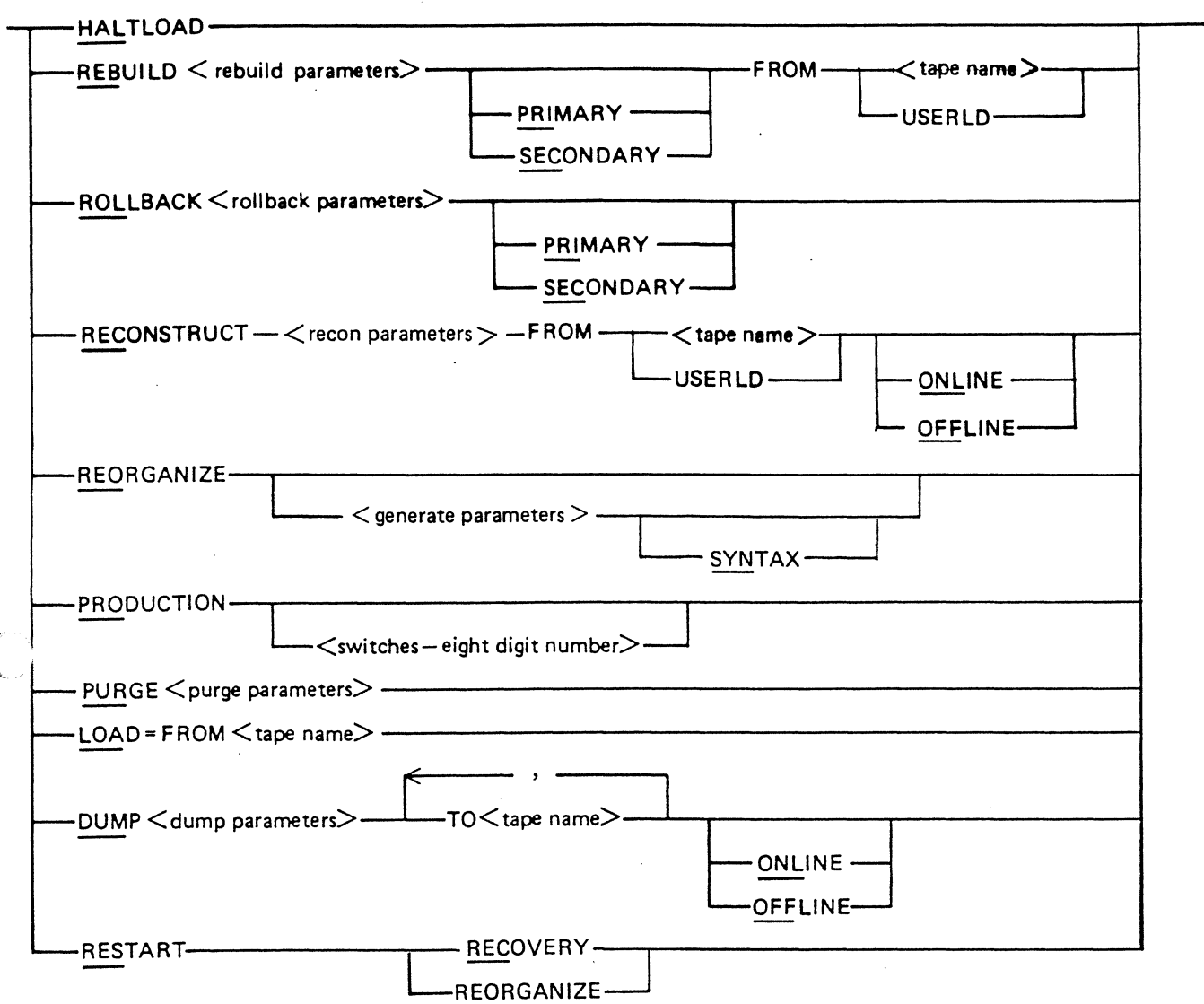
Database name <dbname> is the ID (3 to 6 characters) of the database as specified in the DASDL source. If you specify the PRODUCTION option, the database name must instead be the name of the Production DBP.

The ON DISK, ON PACK, or ON <familyname> clause specifies the location of the Data Definition File (DDF). Use it only with the IR DUMP command.



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Administrative Control Interfaces

<options>



<rebuild parameters>

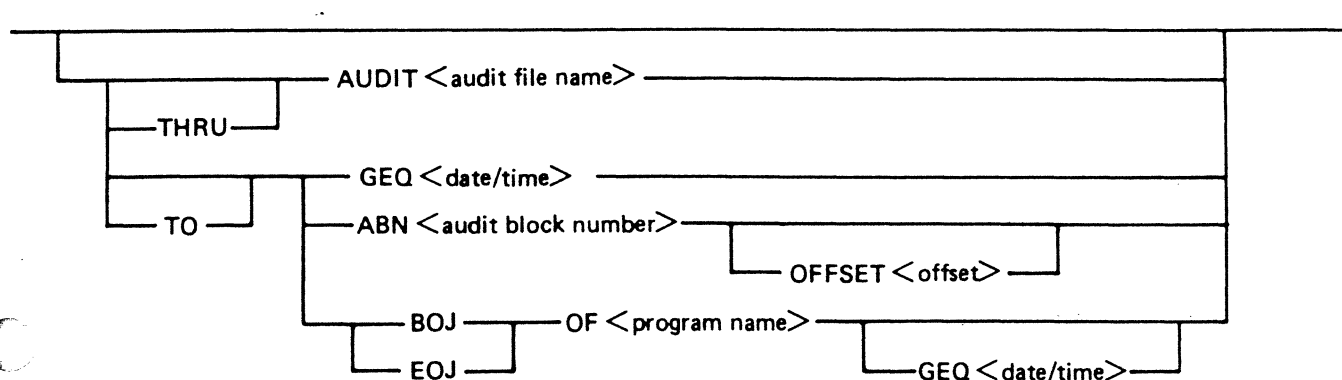
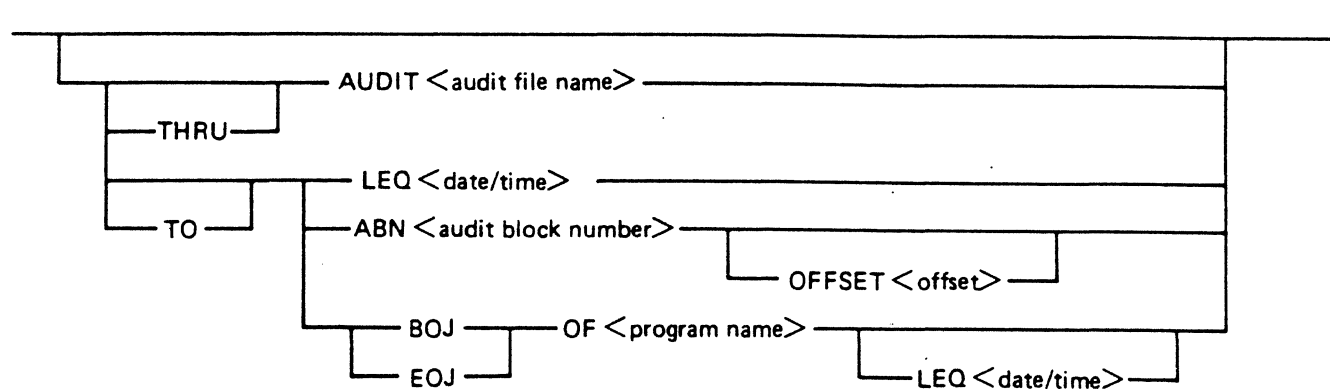


Figure 2-3. IR Command Options Syntax (Sheet 1 of 3)

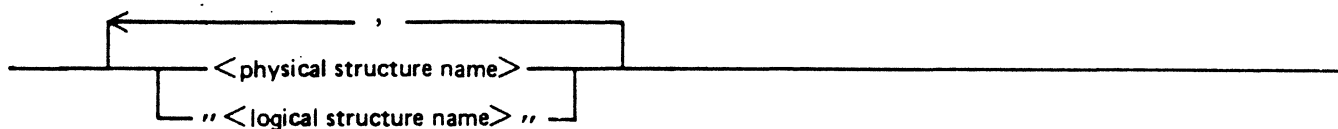


B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Administrative Control Interfaces

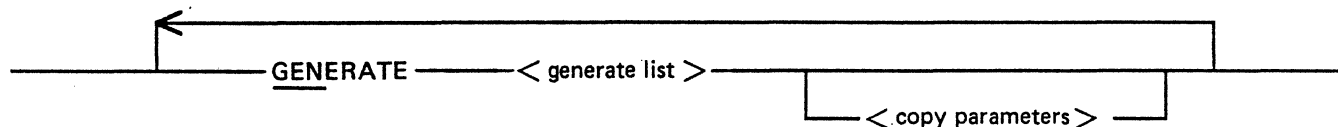
<rollback parameters>



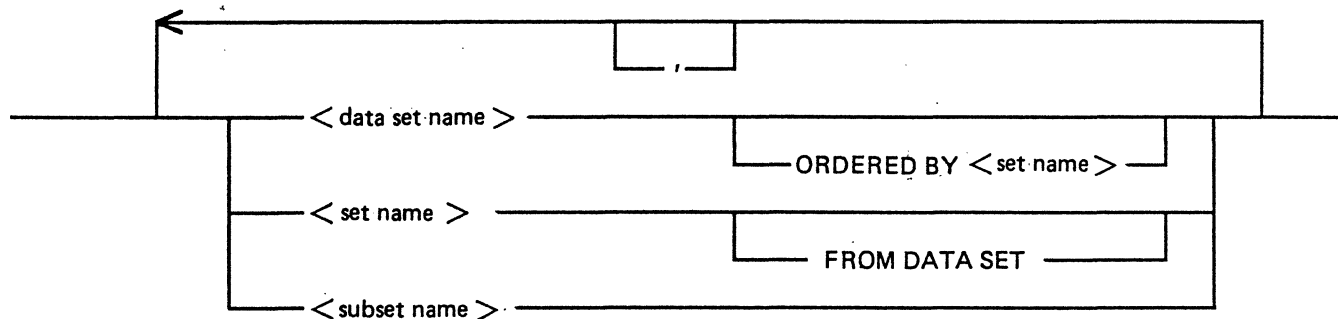
<recon parameters>



<generate parameters>



<generate list>



<copy parameters>

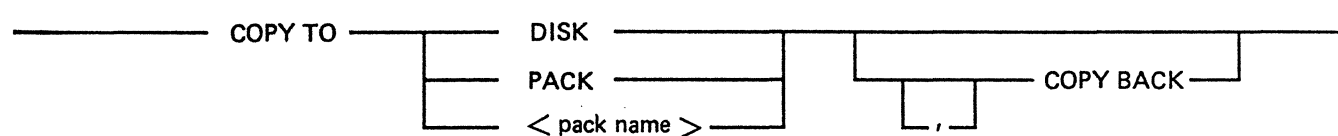


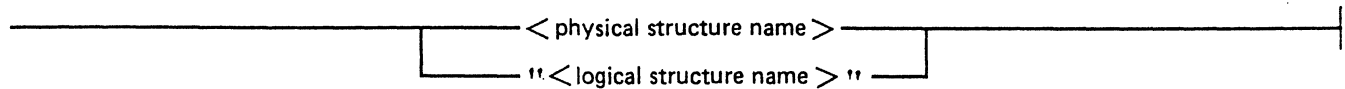
Figure 2-3. IR Command Options Syntax (Sheet 2 of 3)



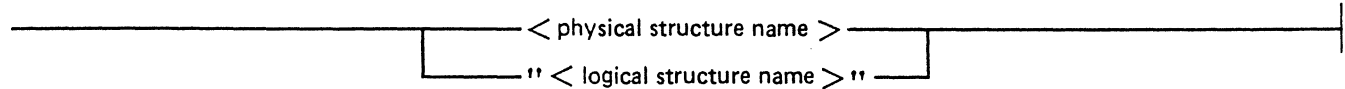


B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Administrative Control Interfaces

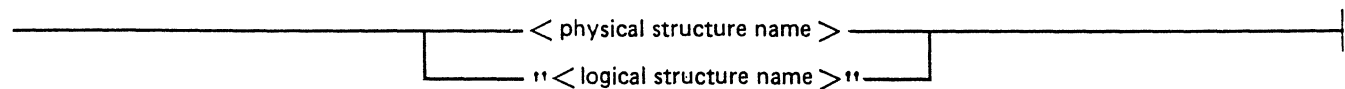
< data set name >



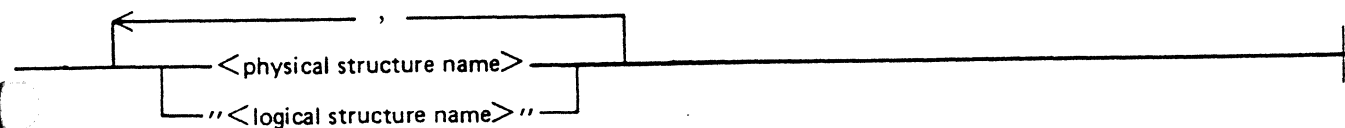
< set name >



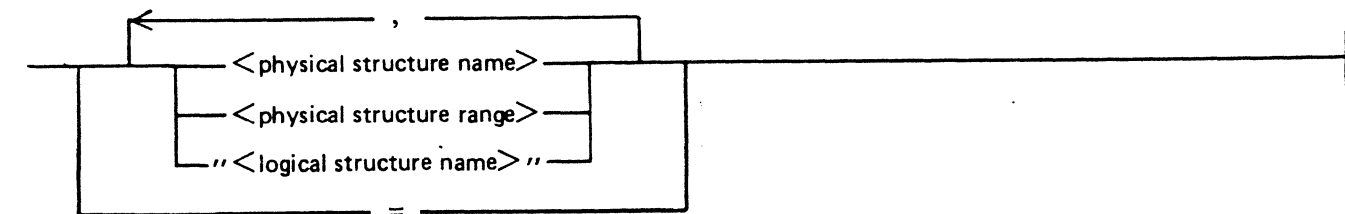
< subset name >



< purge parameters >



< dump parameters >



< date/time >

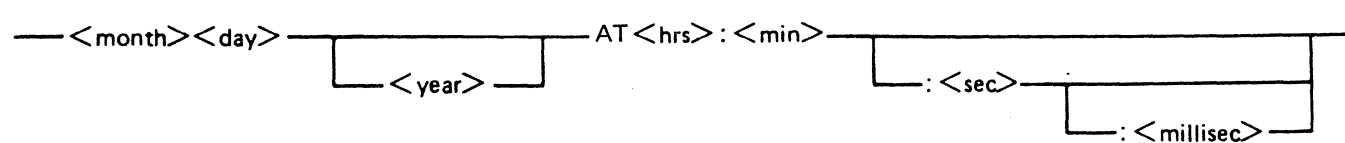


Figure 2-3. IR Command Options Syntax (Sheet 3 of 3)



|                            |                                                                                                                      |
|----------------------------|----------------------------------------------------------------------------------------------------------------------|
| <program name>             | Program ID of program opening database (1 to 6 characters).                                                          |
| <month>                    | JAN, FEB, or JANUARY, FEBRUARY, etc.                                                                                 |
| <day>                      | Day of month. Up to 2 digits.                                                                                        |
| <year>                     | Up to 4 digits.                                                                                                      |
| <hrs>                      | Up to 2 digits.                                                                                                      |
| <min>                      | Up to 2 digits.                                                                                                      |
| <sec>                      | Up to 2 digits.                                                                                                      |
| <millisec>                 | Up to 3 digits.                                                                                                      |
| <audit block number>       | Up to 12 digits.                                                                                                     |
| <physical structure name>  | File name of physical structure file.<br>6 characters in form XXXnnn.                                                |
| <physical structure range> | XXXMMM-XXXNNN, where XXXMMM and<br>XXXNNN are structure names and MMM < NNN. The<br>names are separated by a hyphen. |
| <logical structure name>   | Structure name as specified in DASDL source. 1 to 17<br>characters.                                                  |

## HALTLOAD Option

Specify the HALTLOAD option if a halt/load recovery is to be performed on the database. Note that you do not generally have to enter an IR HALTLOAD command because halt/load recovery runs automatically whenever any program opens the database and the DBP determines that a halt/load recovery is required. Specify the HALTLOAD option only for audited databases.

## REBUILD Option

When you specify the REBUILD option, the recovery process moves the entire database forward in time. Rebuild recovery uses a complete backup of the database and applies after images from the audit trail to bring the database forward to the specified point. Specify the REBUILD option only for audited databases.

The complete database backup must be one of the following:

- A full database dump created with the IR DUMP command, DMSUTL utility; or the SYSTEM/COPY utility.
- A full and complete database backup you have created. This type of backup can be on the media of your choice, but each file must be restored to the media of its origin before the REBUILD recovery. (Use the USERLD syntax of the REBUILD parameters with this type of backup.)

The rebuild process loads the DBP and XXXICM files, if they are not present on the disk. Section 5 contains a discussion of the use of database backups.

If the REBUILD recovery does not complete successfully for any reason (other than a software error or an audit file error), you can restart it with the IR RESTART command. It is not necessary to reenter the REBUILD option.

You can use rebuild recovery to recover from data corruption or to correct logical errors such as those caused by a user program that updates the database with incorrect data.



The rebuild parameters specify the stopping point for rebuild recovery. You specify the stopping point in terms of audit files, the Beginning Of Job (BOJ) or End Of Job (EOJ) of a specific user program, a specific date and time, or a specific audit block number and optionally an offset within the block. If you do not specify a stopping point, rebuild recovery will continue to the end of the audit trail. Because there is no way to determine the last file in the audit trail, you must use the OF MCP command to terminate the recovery. Enter the OF MCP command when the DBP waits on a NO FILE condition because it expects an audit trail file that does not exist.

REBUILD THRU AUDIT <audit file name> applies the after images from audit files created since the time of the dump. This process continues until all after images in the audit file you specify have been applied. Refer to Section 1 for more information on audit file naming conventions.

REBUILD TO BOJ or EOJ applies after images since the dump and stops at the last syncpoint before the BOJ or EOJ of the specified user program.

REBUILD TO GEQ <date/time> applies after images since the dump and stops at the first control record in the audit trail that has a date-timestamp greater than or equal to the specified date/time.

If the date/time falls during a period when a PURGE was in progress, the rebuild continues through the "Purge Done" audit record (for more information about Purging, refer to Section 3).

If the <date/time> falls during a period when an online dump was in progress, the corresponding database dump tape will be invalidated for any future use.

The last step in a rebuild recovery is a halt/load recovery. The halt/load recovery begins by backing the database up to the most recent point in time when no user programs were in transaction state. Note that this may cause transactions to be discarded unless the last record in the audit trail is a syncpoint or equivalent (controlpoint, database CLOSE, etc.).

For example, consider a database for which the last audit is DBXB84. If you specify rebuild recovery through DBXB81, then any old copies of audit files DBXB82 through DBXB84 are invalid. Auditing will begin at the start of audit file DBXB82 when normal updating resumes. It is your responsibility to discard any invalid audit files. If DBXB83 is on disk when the DBP attempts to open a new DBXB83, the DBP will hang waiting on a duplicate file condition. You must then remove DBXB83 from disk or diskpack (RM command) and OK the DBP (OK command). For more details about the RM and OK commands, refer to the *V Series MCP/VS System Software Operations Guide Volume 2: System Commands*.

## ROLLBACK Option

When you specify the ROLLBACK option, the recovery process moves the entire database backward in time. Rollback recovery begins with the current database files. Halt/load recovery occurs, if necessary; then recovery applies audit trail before images to move the database back to the stopping point. You can specify the ROLLBACK option only for audited databases.

Unlike rebuild recovery, rollback recovery cannot fix a data storage corruption problem. However, rollback recovery can save time compared to rebuild recovery when you need to correct logical errors caused by a user program that performed an update with incorrect data.

You specify the stopping point for rollback recovery. The stopping point can be an audit file, the beginning or end of a specific user program, a specific date and time, or a specific audit block number and offset. If you do not specify a stopping point or the specified stopping point does not exist in the audit trail, rollback recovery will continue to the beginning of the audit trail.



Note: You should insure that the stopping point for the rollback recovery exists in the audit trail. If rollback recovery does not find the specified stopping point, it will continue to ask for previous audit files until it has rolled back through audit file XXXA01. If the specified stopping point is not found, rollback recovery will eventually wait on a nonexistent audit file. At this point the only recourse available is to perform a rebuild recovery from a previous database dump. You cannot use the OF MCP command to effect early termination of a rollback recovery.

If the <date/time> falls during a period when an online dump was in progress, the corresponding database dump is not valid for any future use.

Whenever the DBP finds a recovery record in the audit trail, it skips all audit trail records until it finds the previous syncpoint record. In other words, any audit trail records backed out by a previous recovery process are not applied by the rollback recovery.

After finishing each audit file, the DBP displays a message on the ODT, flushes all of the data buffers to disk or diskpack, and updates the control file to reflect the current stable spot in the audit trail. This enables the DBP to restart rollback recovery in the event of a hardware or software failure.

If a rollback recovery encounters a structure discontinuity record in the audit trail, the DBP compares the discontinuity timestamps. If equal, the DBP marks the control file to show that there is a discontinuity in that structure. If the DBP later encounters any more audit records for that structure, it returns a discontinuity error and aborts the rollback recovery.

When the DBP reaches the stopping point in a rollback recovery, it performs a halt/load recovery before terminating. Thus, the DBP backs out any audit records between the stopping point record and the previous syncpoint record. If the stopping point occurs in the middle of an audit file, the DBP updates the End Of File (EOF) for that audit file to reflect the new EOF.

Because of the halt/load recovery that occurs at the end of rollback recovery, you cannot use the OF MCP command to terminate rollback recovery prematurely. Because a rollback recovery backs up to the last syncpoint, the recovery process requires prior audit files. If a rollback recovery is part of an audit file option, the recovery process needs the prior file for backup to the prior syncpoint to guarantee that the database is at a stable point.

The recovery process creates a new audit file which is one greater than the audit file that contains the stopping point. This new audit file contains the rollback record signifying that a rollback recovery has run. Subsequent OPENS of the database use this new file for auditing. You should discard any audit files greater than this new file.

#### NOTE

Take care not to confuse the discarded audit files with the newly created audit files.





## RECONSTRUCT Option

Use the RECONSTRUCT option to recover damaged database files. Reconstruct is a form of rebuild recovery in which you recover only a portion of the database files.

The DBP loads the structures to be reconstructed from a database backup and then brings them up to date with the rest of the database by applying after images from the specified audit trail. The DBP obtains the starting point in the audit trail from the control file (XXXCNO) that is part of the database backup.

Reconstruct recovery can use any of the following database backups:

- A full or partial, online or offline database dump that you created with the IR DUMP command, DMSUTL DUMP, or with the SYSTEM/COPY utility.
- A full or partial offline database backup that you have created. This type of backup can be on the media of your choice, but must be restored to the media of its origin before reconstruction takes place. (Use the USERLD syntax of the RECONSTRUCT parameter with this type of backup.)

Section 5 contains a discussion about the use of database backups that includes the naming of the XXXCNO "old" control file.

Use the ONLINE option to select online reconstruct, during which a user program can access all structures that are not being reconstructed. Use the OFFLINE option to select offline reconstruct, during which user programs are not allowed to use the database. The OFFLINE option is the default.

If the reconstruct recovery fails, you can begin it again by entering the command IR RESTART or by using the RESTART option of the DMSUTL utility.

The reconstruct list consists of physical structure names or logical structure names of structures you want to recover. A physical structure name is the 6-character file ID of a physical file. Refer to Section 1 for more information on file naming conventions. A logical structure name is the 1- to 17-character name given the structure in the DASDL source. The physical structure range is in the form XXXMMM-XXXNNN where XXX represents the first three characters of the database name, MMM and NNN are structure numbers, and MMM < NNN. Separate the names with a hyphen.

## REORGANIZE Option

You use the REORGANIZE option to perform maintenance on or to make physical changes to database structure files. Two approaches to this process are possible.

In the first approach, you enter an IR REORGANIZE command with <generate parameters>. This initiates the generalized REORG UTILITY (DMSREO) to make the changes to the database. Garbage collection of any type of structure, set balancing/generate-new, and subset balancing can be done using this approach. (Generate-new reads the associated data set, creates a new set, and discards the existing set.) Note that structures associated with those that are specified in the generate list may also be implicitly generated as a part of the reorganization process.

If you specify ORDERED BY <set name> for a data set in the generate list, the generated data set will be ordered according to that set. Set name must be an existing set of that data set. Use this option to optimize performance when user programs access the data set primarily by the specified set.



If you do not specify **ORDERED BY <set name>** for a data set in the generate list, the generated data set will be in the same order as the original. In either case, the generation of a data set performs garbage collection on that structure.

If you specify **FROM DATA SET** for a set in the generate list, a new set will be generated from the associated data set and the original set will be discarded. This corresponds to a generate-new on that structure. You can generate only disjoint sets in this manner. You typically use this generation approach when the existing set is corrupted. Note that the order of any duplicates in the original set is lost when you use this generation approach.

If you do not specify **FROM DATA SET** for a set in the generate list, a replacement set will be generated from the original. This action corresponds to a garbage collection and balancing on that structure.

You cannot specify **FROM DATA SET** for a subset using this reorganization approach. The **WHERE** clause associated with the subset is not known by DMSREO, so it cannot generate the replacement subset from the data set. The second approach (**REORG DBP**), which is described in this section, does support this capability.

If you specify **COPY TO DISK/PACK/<pack name>** for the structures in the generate list, you are requesting that the replacement structures be generated on a temporary medium that is different from the medium that the original structures reside on. This is typically done when space on the original medium is limited.

If you do not specify **COPY TO DISK/PACK/<pack name>**, the replacement structures will be generated on the same medium as the original structures (original medium). Implicitly generated structures are usually fixed up directly (that is, a replacement structure is not generated), so this approach is normally used for them also.

If you specify **COPY BACK**, each replacement structure will be copied back to its original medium immediately after it is generated. Otherwise, all **REORG** actions will be allowed to complete first, and then all the structures will be copied back to their original medium(s).

If you specify **SYNTAX**, the generate parameters will be syntactically checked and DMSREO will produce a reorganization summary for your review, but no reorganize actions will actually be performed. Then, you can use the **RESTART** option at a later time to initiate the previously specified reorganize actions.

The second reorganization approach requires that you modify and recompile the DASDL source to describe the desired changes and bind a tailored **REORG DBP (XXXREO)**. You then initiate the **REORG DBP** to make the changes to the database by entering **IR REORGANIZE** (without any <generate parameters>). Record format changes, file attribute changes, garbage collection, set balancing/generate-new, and subset balancing/generate-new can be done using this approach.

For more information on this approach to reorganization, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*.

If a reorganization process does not complete successfully, you can restart it with the **IR RESTART** command. For more information, refer to the **RESTART** option described in this section.



It is important to recognize that the reorganization process causes structure discontinuities in the audit trail of an audited database. This causes the following recovery restrictions:

- Rebuild recovery cannot proceed through a reorganize discontinuity audit record. Therefore, you may not use a dump created before the reorganization to rebuild the database to a point after the reorganization.
- Reconstruct recovery cannot proceed through a reorganize discontinuity audit record if any of the structures being reconstructed were included in the reorganization. Therefore, you may not use a dump created before the reorganization to reconstruct any structures involved in the reorganization.
- Rollback recovery can proceed through a reorganize discontinuity audit record, but the rollback will be aborted if any update audit records corresponding to any of the structures involved in the reorganization are encountered before the stopping point of the rollback is reached.

#### NOTE

You should dump all of the structure files involved in the reorganization (including implicitly generated structures), the control file, the DDF, the memory file, the tailored code file (ICM), and the DBP both before and after each reorganization. If the reorganization does not complete successfully or must be restarted, you will need the dump you made before the reorganization. You will need the dump you created after the reorganization if a rebuild recovery or a reconstruct recovery that involves any of the reorganized structures is necessary.

For more information on the reorganization process, refer to Section 3.

## PRODUCTION Option

Use the PRODUCTION option to initiate the user program code bound into a Production DBP. The database name you use is that of the Production DBP. Refer to Section 6 for more information.

If a user program other than the bound-in user program opens the database, the Production DBP starts automatically but the bound-in user program code does not execute. In this case, you do not need an IR command. If you want the bound-in user program to be initiated when the Production DBP is already executing, use an explicit IR PRODUCTION command. You can use the DMSUTL utility or the DMSDLR utility instead of the IR command.

If the user program that is bound in with the Production DBP uses program switches, you can set the switches at execution time by following the IR command with an INSERT or a VALUE MCP control instruction. Refer to the *V Series MCP/VS System Software Operations Guide Volume 1: Installation* for a discussion of switches and the INSERT and VALUE commands. An example of setting switches follows.

```
IR MYDB PRODUCTION;INSERT 0 8 01100010
```



## PURGE Option

Use the PURGE option to delete all data records from the specified structure(s). All data is lost from the structure and the structure is reduced. You can do a PURGE only for data sets and manual subsets, both disjoint and embedded. If a structure has to be emptied regularly, a PURGE is quicker than clearing the structure record by record with a user program. Note: You cannot do a PURGE when a user program is accessing the database. For more information about the PURGE process, refer to Section 3.

## LOAD Option

Use the LOAD option to load a backup tape containing information previously dumped with the IR DUMP command, the DMSUTL utility, or the SYSTEM/COPY utility. Refer to Section 5 for a discussion of the use of database backups.

If files of the same names as the loaded files already exist on disk or diskpack, the loaded files will overwrite the existing files.

All of the structure files on the tape will be loaded; use the IR RECONSTRUCT command with the option FROM <tape name> to selectively load some of the structure files.

With the LOAD option, you can load only a full database backup of a stable database. The backup must have been created with an offline dump and must not require recovery. A discussion of database dumping and loading appears in Section 5.

If the LOAD procedure fails, you can enter the IR RESTART command to restart.

## DUMP Option

Use the DUMP option to dump the information in a database to archive tape(s). This function will dump all database files, except audit files. Refer to Section 5 for a discussion of dumping and loading of backup tapes.

The ON DISK, ON PACK, or ON <familyname> clause with the DUMP option specifies the current location of the DDF. The default value is ON PACK.

TO <tape name> specifies the name of the backup tape that the DUMP function creates. If you enter multiple tape names, separated with commas, duplicate tapes are created.

The ONLINE and OFFLINE options specify online dumping and offline dumping, respectively. A discussion of these forms of dumping appears in Section 5. The default value is OFFLINE.

An equal sign (=) for the dump list produces a full database dump. A list of structures or structure ranges produces a partial database dump. Even if you enter the names of all of the structure files, a partial dump will be created. It will contain the control file, but not the other files listed in Figure 5-1. Refer to Section 5 for details.

A physical structure name is the 6-character file name of a physical file. A logical structure name is the 1- to 17-character name given to the structure in the DASDL source. The physical structure range is in the form XXXMMM - XXXNNN where XXX represents the first three characters of the database name, MMM and NNN are structure numbers, and MMM < NNN. You must separate the names with a hyphen.





## RESTART Option

Use the RESTART option to restart certain functions that have not completed because of a nonerror failure. You may also use the RESTART option to initiate a reorganization that you previously entered as a SYNTAX run. You can use restart with the following functions:

- REBUILD
- RECONSTRUCT
- LOAD
- REORGANIZE

Each of these functions proceeds through two or more phases during its execution. A failed function will be restarted in or before the phase it was in when the failure occurred. You do not have to reenter the syntax of the IR command that initiated the failed function.

When you enter the restart command, the syntax of the IR command that initiated the failed function will be displayed on the ODT along with some instructions and a list of action choices.

Enter RESTART RECOVERY to restart a failed rebuild recovery, reconstruct recovery, or database load. After DMSDLR displays the IR syntax and some instructions, you must enter one of the following with an ACCEPT command (for example, 12 AX RES where 12 is DMSDLR's mix number):

- RES - to restart the failed function.
- REM - to remove the database's memory of the failed function and go to EOJ. Note that this choice will probably leave the database in an inconsistent state because the partial results of the failed function are not backed out.
- EXI - to exit from the restart function without doing anything.

Enter RESTART REORGANIZE to restart a failed reorganize function. After DMSDLR displays the IR syntax and some instructions, you may have to reload some of the structures from the dump that was taken before the reorganization was initiated (this will be explained in the instructions). Then, you must enter one of the following with an ACCEPT command (for example, 12 AX RES where 12 is DMSDLR's mix number):

- RES - to restart the failed reorganization or to initiate a reorganization that you entered previously as a SYNTAX run.
- REM - to remove the database's memory of the failed reorganization and go to EOJ. Note that this choice will probably leave the database in an inconsistent state because the partial results of the failed reorganization are not backed out. You should then reload all of the files from the dump you made before you initiated the reorganization.
- EXI - to exit from the restart function without doing anything.
- SYN - to perform another SYNTAX run (and create another reorganization summary). This choice is only offered if you did a SYNTAX run originally.
- RGO - to do the same thing as REM.

If the DBP detects (at BOJ) that a previous restartable function has failed, it will display an error indicating this and go to EOJ. If this occurs, you should then use the RESTART function.



A notable exception to this is the REORGANIZATION function. If you initiate a reorganization and DMSDLR detects that a previous reorganization has failed, it will first perform a restart reorganize. You will be offered the same action choices as for restart reorganize, except for the following:

RGO - to remove the database's memory of the failed reorganization and proceed with the newly entered reorganization.

## SB Command

To obtain information about database activity, you can enter an SB command conforming to the syntax that appears in Figure 2-4.

<SB command>

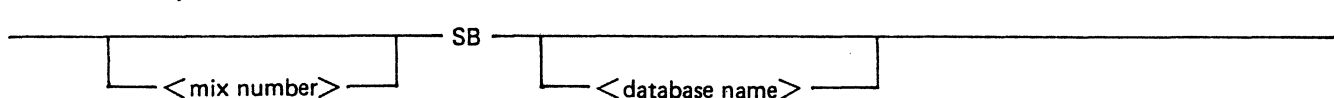


Figure 2-4. Database Activity (SB) Command Syntax

If you specify the mix number of a user program with the SB command, a list of the active databases associated with that program appears on your terminal. If you specify a database name with the SB command, a list of all user programs currently accessing that database appears on your terminal. If you specify neither a mix number nor database name, a list of all active databases appears on your terminal. You should not specify both a mix number and a database name because the presence of one excludes the use of the other.

Examples:

|         |                                                                    |
|---------|--------------------------------------------------------------------|
| Input:  | 12 SB                                                              |
| Output: | P=7 M=7 DMS/DATAB1=32 WTG DMS FUNCTION                             |
| Input:  | SB TESTDB                                                          |
| Output: | P=4 M=4 PROGD1=20 EXECUTING<br>P=4 M=4 TESTV1 =05 WTG DMS FUNCTION |
| Input:  | SB                                                                 |
| Output: | SB= DATAB1 TESTDB DBACC5                                           |



## SP Command

Use the SP command, in the format that appears in Figure 2-5, to interrogate the DBP as to the value of certain parameters and to set the values of certain parameters while the DBP is executing.

<SP command>

— <mix number> SP —

---

Figure 2-5. SP Command Syntax

<mix number> is the mix number of the DBP.

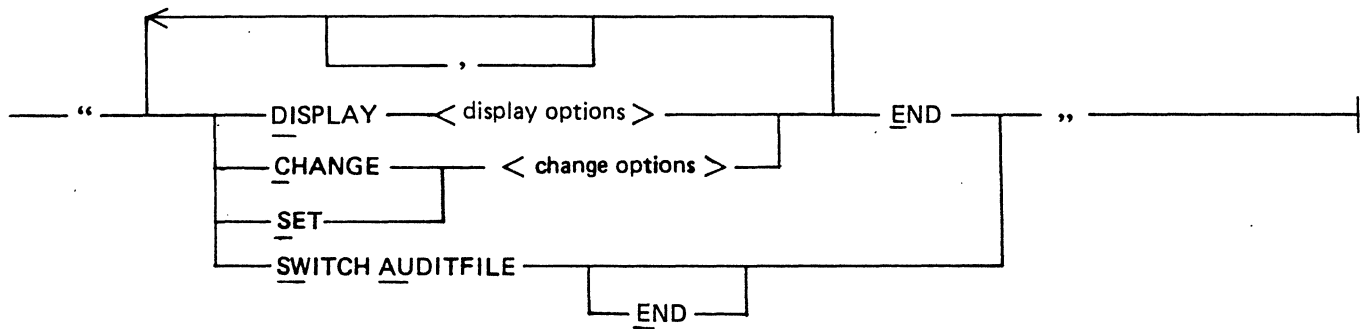
### NOTE

The SP command does not work when the only user program executing is bound into a Production DBP. If other user programs are accessing the database at the same time the Production DBP is running, the SP command will work.

When you enter the SP command, the DBP awaits parameters that you enter with the AX (ACCEPT) command. The parameters you enter must be in the form that appears in Figure 2-6. You can enter just the letters that are underlined or any combination of letters including those that are underlined.



< SP parameters >



< display options >

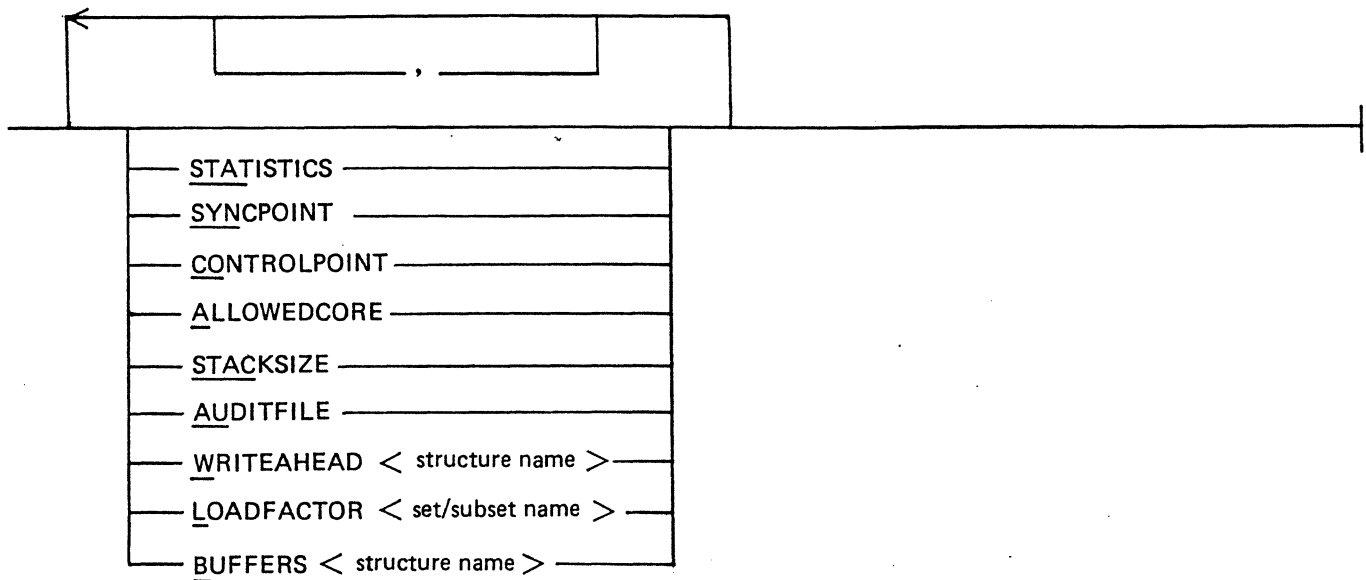


Figure 2-6. SP Command Parameters (Sheet 1 of 2)





< change options >

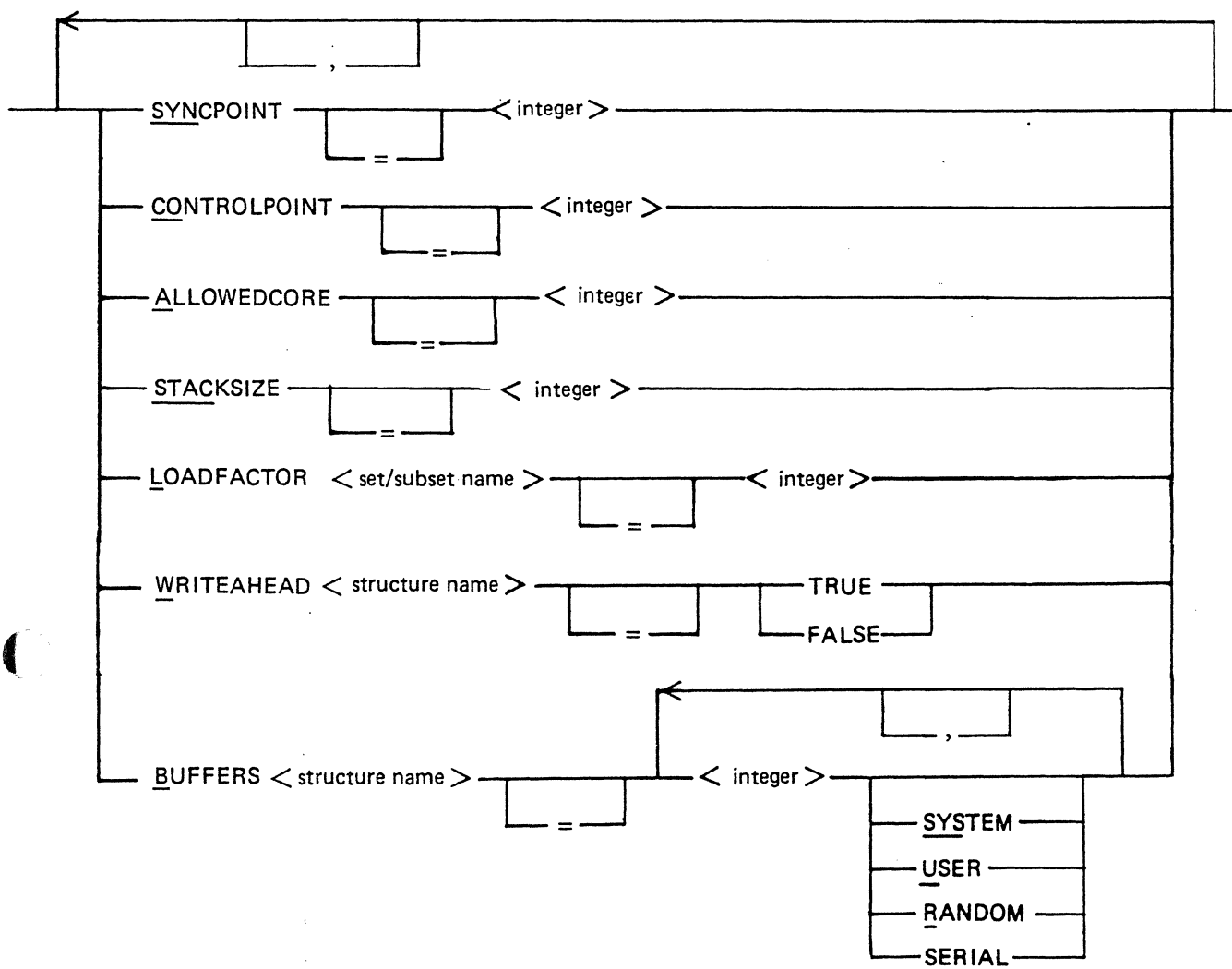


Figure 2-6. SP Command Parameters (Sheet 2 of 2)

If you do not enter either DISPLAY, SET, SWITCH or CHANGE, the default value is DISPLAY.

Use SET to update the value of certain parameters for one execution of the DBP only.

Use CHANGE to update the value of parameters permanently or until a DASDL compilation changes them.



Use SWITCH AUDITFILE to instruct the DBP to close the database's current audit trail file and open the next sequential audit trail file. This option of the SP command lets you back up or verify the audit files without bringing down the DBP. The SWITCH AUDITFILE option is helpful in audit file backup or verification procedures. When it gets this command, the DBP writes an audit file change record in the new audit file at the next syncpoint. This forces a double controlpoint.

Use [DISPLAY] STATISTICS to find out whether the STATISTICS option is SET or RESET. You cannot change the value of the STATISTICS option with the SP command.

Use [DISPLAY] SYNCPOINT to find out the number of transactions per syncpoint. Use SET or CHANGE SYNCPOINT = <integer> to change the value of the syncpoint parameter. The integer you specify must be between 1 and 9999 inclusive.

Use [DISPLAY] CONTROLPOINT to find out the number of syncpoints per controlpoint. Use SET or CHANGE CONTROLPOINT = <integer> to change the value of the controlpoint parameter. The integer you specify must be between 1 and 9999 inclusive.

Use [DISPLAY] ALLOWEDCORE to find out the maximum amount of total memory that the DBP can expand to. This includes base memory and dynamic memory. Use SET or CHANGE ALLOWEDCORE = <integer> to change the value of ALLOWEDCORE to the integer you specify. If ALLOWEDCORE has a beginning value less than or equal to one million digits (1MD), and the new value is greater than one million digits, a SET command will not be allowed and a CHANGE will only affect the value for any subsequent runs of the database program. The current value will not be changed. Refer to the ALLOWEDCORE core option in the database description that appears in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*.

Use [DISPLAY] STACKSIZE to find out the amount of total memory (in digits) that the DBP uses for static memory. This is the smallest amount of memory that can be assigned to the DBP. Use SET or CHANGE STACKSIZE = <integer> to change the value of STACKSIZE to the specified number.

Use [DISPLAY] AUDITFILE to determine if the AUDITFILE option is SET or RESET. If the AUDITFILE option is SET, the DBP displays the names of the current primary and secondary audit files along with the current audit block number. You cannot change the value of the AUDITFILE option with the SP command.

Use [DISPLAY] LOADFACTOR <set/subset name> to find out the LOADFACTOR percentage of the set or subset. LOADFACTOR determines the percentage of entries kept in a table when it is full and must be split. For example, 50 = 50% and 80 = 80%. In other words, an 80% LOADFACTOR indicates that the two new tables will be 80% full. The maximum allowable LOADFACTOR is 99. The default value is 80. DISPLAY LOADFACTOR shows the number of entries that the percentage represents. Use SET or CHANGE LOADFACTOR = <integer> to change the value of LOADFACTOR to the number you specify.

Use [DISPLAY] BUFFERS <structure name> to find out the number of buffers currently allowed for the <structure name>. <structure name> is the 1- to 17-character structure name you specified in the DASDL input.



Use SET or CHANGE BUFFERS <structure name> = <integer> [SYSTEM] to change the number of system buffers for the structure name. The maximum value that you can enter as an integer is 99, but the maximum limit for all buffers for a database is also 99. The sum of the number of system buffers, user buffers (also called random buffers), and serial buffers cannot exceed 99.

Use SET or CHANGE BUFFERS <structure name> = <integer> USER to change the number of buffers that will be added for each user program that accesses the database randomly. "User" and "random" buffers are synonyms. The sum of the number of system buffers, user buffers (also called random buffers), and serial buffers cannot exceed 99.

Use SET or CHANGE BUFFERS <structure name> = <integer> SERIAL to change the number of buffers that will be added for each user program that accesses the database serially. The sum of the number of system buffers, user buffers (also called random buffers), and serial buffers cannot exceed 99. You activate serial optimization (readahead) by setting this value greater than or equal to 2.

Use SET WRITEAHEAD <structure name> = TRUE to cause the buffers of a structure you specify to be written before they must be written in response to block requests. A complete discussion of WRITEAHEAD appears in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation* under the discussion of the Data Set Physical Option specification.

Use WRITEAHEAD <structure name> = FALSE to reset the WRITEAHEAD option for the named structure. With WRITEAHEAD RESET, the DBP writes buffers only when the buffer is needed for another block. In rare cases, WRITEAHEAD can cause extra writes. In these cases, performance can be improved by not using WRITEAHEAD.

#### NOTE

By default, DASDL sets WRITEAHEAD to TRUE for every structure.

You must specify END as the last command. The DBP continues to expect ACCEPT messages until you enter END.



## SECTION 3

# UPDATE, REORGANIZATION, AND PURGING

### INTRODUCTION

During the life of the database, the demands upon it may change. Use the update and reorganization functions to modify the database to meet these new demands.

Use the update function to make modifications to the database that do not result in changes to the physical format of files or records. Use the reorganization function when physical changes to the format of database files or records are necessary.

### UPDATE

Use the update function in the following instances:

- To change options, parameters, and physical attributes.
- To add and delete disjoint data sets and, in specific instances, sets and subsets.
- To add or delete embedded data sets and sets, and to add or delete the owner data set.
- To convert an existing database to a newer release of the Data Management System II (DMSII) software. For conversion, you cannot make any changes to the DASDL source except to add the \$UPDATE statement.

An update of any audit trail attribute, except CHECKSUM and UPDATE-EOF (End Of File), results in a discontinuity in the audit trail. It is not possible to recover a database across this discontinuity. Therefore, care must be exercised in changing audit trail attributes. The integrity of the database should not be in question, and the database should be backed-up to another medium before and after making the update.

### Performing Update

To perform updates, make the necessary changes to the original DASDL source. Recompile the modified DASDL source. The tables at the end of this section indicate which changes are allowed with the update function. Figure 3-1 shows a diagram of an update compile.





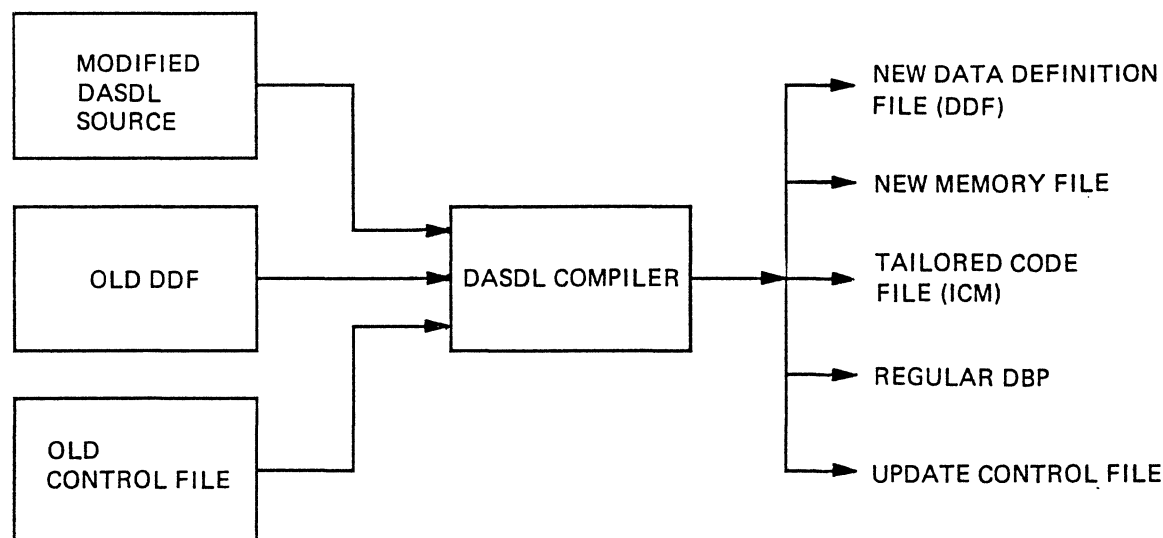


Figure 3-1. Update Compile

The \$UPDATE option instructs the DASDL compiler to update the existing database description. Insert \$UPDATE near the beginning of the DASDL source, before the existing database description. The line containing \$UPDATE must precede the first line that does not contain a \$ option.

Declare disjoint data sets, sets, and subsets in any order; they do not have to be in the same order as in the original DASDL source. Data items, however, must remain in the original order within a record because the order affects record format.

Following a syntactically correct compilation, the DASDL compiler compares the new database description to the old description. If all changes are valid, the DASDL compiler transfers the date/timestamp from the old description to the new description so that existing user programs can run without recompilation.

The DASDL compiler then creates a new database program tailored code file and binds the database program (DBP). The name of the new tailored code file is XXXICM, where XXX represents the first three letters of the database name.

The deletion of data sets or sets causes a discontinuity in the database. If the database is audited, the DBP makes a record of this discontinuity in the audit trail. Database recovery cannot backup past or roll forward through this discontinuity.

If you add a data set or set to the database, the structure number the DASDL compiler assigns is the next highest structure number available or the structure number of a deleted structure from the last update run, if any.

Generally, user programs need not be recompiled following the update process; however, some programs may require modification. For example, if you delete a data set, you must eliminate all references to the data set from existing user programs.



Example 1: Changing data set buffers and set buffers

Original DASDL Source:

```
$ SET LIST
D DATA SET
(
  A ALPHA (10);
  B BOOLEAN;
  N NUMBER (6) OCCURS 15 TIMES;
)
S SET OF D
  KEY IS A;
  .
  .
  .
D(
  BLOCKSIZE =      42 RECORDS,
  AREALENGTH =    2352 RECORDS,
  BUFFERS    =       1 PER USER,
  AREAS      =       5
);
S(
  BLOCKSIZE =     159 ENTRIES,
  AREALENGTH =      48 BLOCKS,
  AREAS      =       3,
  BUFFERS    =    1 + 1 PER USER,
  LOADFACTOR =      66
);
```



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Update, Reorganization, and Purging

---

DASDL Source for Update Compile:

```
$UPDATE                                %added (calls for Update compile)
D DATA SET
(
  A ALPHA (10);
  B BOOLEAN;
  N NUMBER (6) OCCURS 15 TIMES;
)
S SET OF D
  KEY IS A;
  .
  .
  .
D(
  BLOCKSIZE   =      42 RECORDS,
  AREALENGTH  =    2352 RECORDS,
  BUFFERS     =       2 PER USER,      %changed
  AREAS       =       5
);
S(
  BLOCKSIZE   =    159 ENTRIES,
  AREALENGTH  =     48 BLOCKS,
  AREAS       =       3,
  BUFFERS     =    2 + 1 PER USER,     %changed
);
```

In example 1, note that BUFFERS were increased for both the data set and the set. AREAS, AREALENGTH, and BLOCKSIZE were included to make sure that they did not change from the previous compile. LOADFACTOR has been dropped which results in a new LOADFACTOR of 80% (the default value) for S.



Example 2: Changing options, parameters, and audit trail attributes

Original DASDL Source:

```
OPTIONS
(
  AUDIT,
  STATISTICS
);

AUDIT TRAIL ATTRIBUTES
(
  FAMILYNAME = AUDPAK,
  CHECKSUM SET
);

D DATA SET
(
  A ALPHA (5);
  B BOOLEAN OCCURS 5 TIMES;
  R NUMBER (6);
);

RS RESTART DATA SET
(
  X ALPHA (10);
  Y ALPHA (100);
);
```





DASDL Source for Update Compile:

```
$UPDATE                                %added (calls for Update compile)
OPTIONS
(
  AUDIT
);

PARAMETERS                            %added
(                                     %added
  CONTROLPOINT = 10                  %added
  SYNCPOINT = 50                     %added
);                                   %added

AUDIT TRAIL ATTRIBUTES
(
  BLOCKSIZE = 3570 BYTES,            %added
  FAMILYNAME = AUDPACK,
  CHECKSUM SET
);

D DATA SET
(
  A ALPHA (5);
  B BOOLEAN OCCURS 5 TIMES;
  R NUMBER (12);
);
RS RESTART DATA SET
(
  X ALPHA (10);
  Y ALPHA (100);
);
```

In example 2, in addition to the statements that were added, the STATISTICS option was RESET by deleting it from the input.



## REORGANIZATION

Reorganization provides you with a convenient method of making physical changes to the database files and records. The capabilities of reorganization are:

### Database Maintenance

- Consolidates deleted and unused space in data sets. This is called compression or garbage collection.
- Balances sets and subsets by restructuring index tables to achieve a uniform distribution of index entries. You do this to achieve the most efficient access of the data set when you are accessing it by means of a particular set or subset. Garbage collection also occurs when you balance a set or subset.
- Removes “orphan” entries in manual subsets. Orphan entries occur when a user program deletes a data set record without removing the manual subset entry that refers to it.
- Generates new sets to replace existing ones by accessing their associated data sets (generate-new).
- Generates new sets or automatic subsets to access existing data sets.
- Orders the records in a data set according to a set that references it.

### File Format Conversion

- Changes the physical attributes of structure files (for example: number of AREAS, AREALENGTH, BLOCKSIZE, LOADFACTOR, etc.).

### Record Format Conversion

- Changes the physical attributes of data set record items (for example: type, size, scale factor, number of occurrences, etc.).

There are two approaches that you can use to perform a reorganization.

In the first approach, you enter an IR REORGANIZE command with generate parameters (or the equivalent using DMSUTL or EX DMSDLR). This initiates the generalized REORG utility (DMSREO) to make the changes to the database.

In this approach, you do not do a DASDL compile, but you are limited to the capabilities that are listed under Database Maintenance. For more information, see the “Performing Database Maintenance with DMSREO” portion of this section.

In the second approach, there are two phases to the database reorganization: the reorganization compile and the reorganization run.

In the reorganization compile, the DASDL compiler processes the modified DASDL source and GENERATE statements for the structures you are reorganizing. The DASDL compiler then sorts the information for all the reorganized structures and writes it to the memory file. The DASDL compiler creates a new Data Definition File, memory file, and Tailored Code File. It also creates a special Tailored Code File for the reorganization and a special database program to carry out the reorganization of the physical structure files. This special database program is called the Reorg DBP. The DASDL compiler may also create a new regular DBP. When the reorganization compile completes successfully, the DASDL compiler marks the control file to indicate that a reorganization run is required. Then the database cannot be used until the reorganization run is completed.



Figure 3-2 shows a diagram of the reorganization compile.

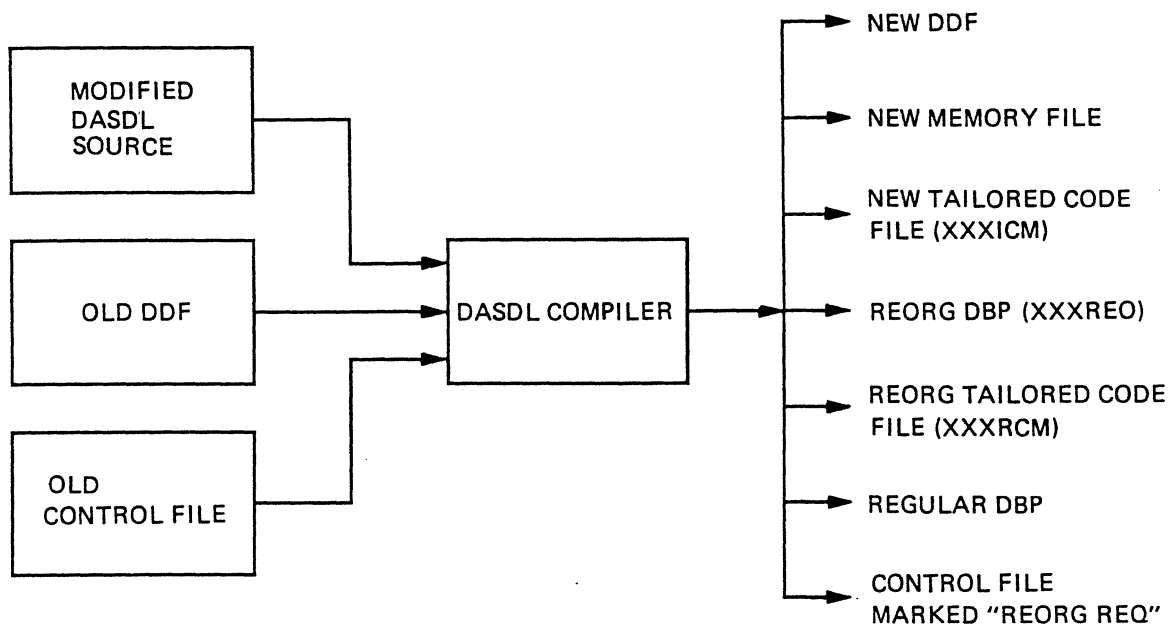


Figure 3-2. Reorganization Compile

You initiate the reorganization run by executing DMSDLR, with the IR command, by putting \$AUTOREORG in the reorganization DASDL compile, or with the DMSUTL utility. The Reorg DBP loads all reorganization structure information from the memory file. The Reorg DBP then reads old database files and creates new database files by doing the appropriate reorganization action on the old structures.

Figure 3-3 shows a diagram of the reorganization run.



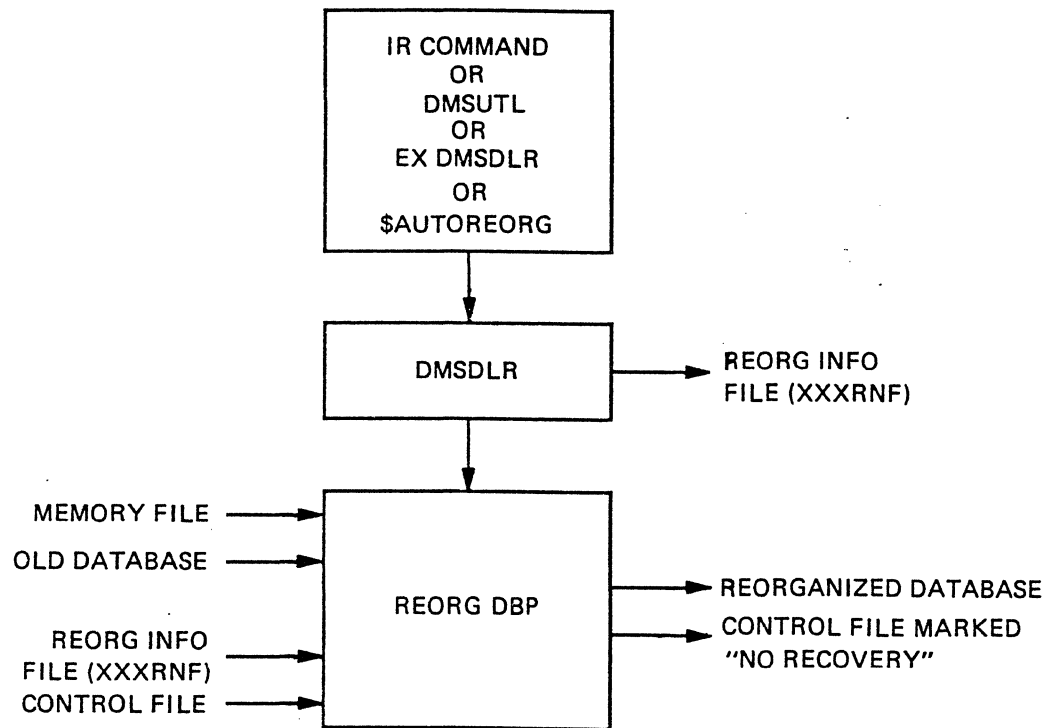


Figure 3-3. Reorganization Phase





## Precautions and Limitations

It is important to recognize that the reorganization process causes structure discontinuities in the audit trail of an audited database. This causes the following recovery restrictions:

- Rebuild recovery cannot proceed through a reorganize discontinuity audit record. Therefore, you may not use a dump created before the reorganization to rebuild the database to a point after the reorganization.
- Reconstruct recovery cannot proceed through a reorganize discontinuity audit record if any of the structures being reconstructed were included in the reorganization. Therefore, you may not use a dump created before the reorganization to reconstruct any structures involved in the reorganization.
- Rollback recovery can proceed through a reorganize discontinuity audit record, but the rollback will be aborted if any update audit records corresponding to any of the structures involved in the reorganization are encountered before the stopping point of the rollback is reached.

### NOTE

You should dump all of the structure files involved in the reorganization (including implicitly generated structures), the control file, the DDF, the memory file, the Tailored Code File (ICM), and the DBP both before and after each reorganization. If the reorganization does not complete successfully or must be restarted, you will need the dump you made before the reorganization. You will need the dump you created after the reorganization if a rebuild recovery or a reconstruct recovery that involves any of the reorganized structures is necessary.

The reorganization process may fail with a LIMITERROR message when it is balancing an indexed sequential set. For example, you have a data set with 99 entries. It is accessed with a set that has these attributes:

- Areas = 2
- Blocks/Area = 4
- Records/Block = 100
- LOADFACTOR = 67 (MAXRECORDS is not specified)

The system reorganizes it as follows:

- Areas = 1
- Blocks/Area = 3
- Records/Block = 102
- LOADFACTOR = 50

The reorganization fails with a LIMITERROR on the set because the DBP needed four blocks to accommodate Block one, the root table and two fine tables, so that it could balance the set by splitting the root table. To avoid the LIMITERROR, you should remember that Block one is reserved.



## Database Maintenance

The database maintenance features of reorganization enable you to consolidate unused space in structures, balance sets and subsets, remove “orphan” manual subset entries (those without corresponding data set records), generate a new set or automatic subset to access an existing data set, generate a new set to replace an existing one by accessing its associated data set (generate-new), and to reorder the records in a data set according to a set that references it.

When garbage collection (compression or balancing) consolidates unused space in data sets, sets, and subsets the structure file shrinks accordingly.

The balancing process handles root, coarse, and fine tables differently. For example, it fills fine tables to LOADFACTOR % full and it fills coarse and root tables to 100% full. In general, it is a good idea to increase the LOADFACTOR of a set or subset during a GENERATE function if it is not frequently updated, because the increase in LOADFACTOR may reduce the amount of disk or diskpack space required by the set or subset.

You can change LOADFACTOR back to its original value after the generate process with the SP command or with a DASDL update.

### NOTE

Reorganization will not fully compress random and direct data sets. Garbage collection of these data sets will only return unused space at the end of the structure.

Compression of a data set may result in an existing set being generated new rather than a fine table fixup if one of the keys is DATA-SET-RRN. This is because of the changing relative record numbers in the data set.

## Performing Database Maintenance with DASDL (XXXREO)

You perform database maintenance by compiling the original DASDL source with a \$REORGANIZE record and a GENERATE statement, if you do not make changes to the database. If you make changes to the database, the DASDL compiler will automatically add the GENERATE statement for the files being changed. Figure 3-4 shows an example of DASDL source.

```
? COMPILE TESTDB WITH DASDL TO LIBRARY
? FILE DDF DBPACK/TESDDF
? DATA CARD
$ SET REORGANIZE
A STANDARD DATA SET
(
  A1 ALPHA (28);
  A2 NUMBER (06);
  A3 BOOLEAN;
);
S SET OF A
  KEY IS A2;
GENERATE A;
? END
```

Figure 3-4. Example of DASDL Source for a Database Maintenance Reorganization



If the DASDL compilation is error-free, the DBP produces two database ICMs (XXXICM and XXXRCM) and two database programs (XXXXXX and XXXREO). A diagram of this process appears in Figure 3-2.

Once the DASDL compilation completes successfully, it marks the database control file to indicate that reorganization is required. No further access may be made to the database until the DBP completes the reorganization process.

You start the reorganization process with the IR REORGANIZE command, described in Section 2; the DMSDLR option of the DMSUTL utility, described in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*; or the \$SET AUTOREORG compiler control record.

## Performing Database Maintenance with DMSREO

The reorganization utility, DMSREO, is a generalized version of the Reorg DBP that is tailored to perform database maintenance without requiring you to recompile the DASDL source. You use the IR REORGANIZE command (with generate parameters), the reorganize option in DMSUTL, or EX DMSDLR to initiate DMSREO to perform the reorganization.

The following description assumes that you are using the IR command syntax, but the DMSUTL screen choices are similar.

You can perform most of the database maintenance capabilities, and you do not have to recompile any existing user programs when you use this reorganization approach.

Garbage collection of any type of structure, set balancing/generate-new, and subset balancing can be done using this approach. (Generate-new reads the associated data set, creates a new set, and discards the existing set.) Note that structures associated with those that are specified in the generate list may also be implicitly generated as a part of the reorganization process.

DMSREO provides three options (ORDERED BY, FROM DATA SET, and COPY TO) with the GENERATE statement.

If you specify ORDERED BY <set name> for a data set in the generate list, the generated data set will be ordered according to that set. Set name must be an existing set of that data set. Use this option to optimize performance when user programs access the data set primarily by the specified set.

If you do not specify ORDERED BY <set name> for a data set in the generate list, the generated data set will be in the same order as the original. In either case, the generation of a data set performs garbage collection on that structure.

If you specify FROM DATA SET for a set in the generate list, a new set will be generated from the associated data set and the original set will be discarded. This corresponds to a generate-new on that structure. You can generate only disjoint sets in this manner. You typically use this generation approach when the existing set is corrupted. Note that the order of any duplicates in the original set is lost when you use this generation approach.

If you do not specify FROM DATA SET for a set in the generate list, a replacement set will be generated from the original. This action corresponds to a garbage collection and balancing on that structure.

You cannot specify FROM DATA SET for a subset using this reorganization approach. The WHERE clause associated with the subset is not known by DMSREO, so it cannot generate the replacement subset from the data set. The second approach (REORG DBP), which is described in this section, does support this capability.



If you specify `COPY TO DISK/PACK/<pack name>` for the structures in the generate list, you are requesting that the replacement structures be generated on a temporary medium that is different from the medium that the original structures reside on. This is typically done when space on the original medium is limited.

If you do not specify `COPY TO DISK/PACK/<pack name>`, the replacement structures will be generated on the same medium as the original structures (original medium). Implicitly generated structures are usually fixed up directly (that is, a replacement structure is not generated), so this approach is normally used for them also.

If you specify `COPY BACK`, each replacement structure will be copied back to its original medium immediately after it is generated. Otherwise, all REORG actions will be allowed to complete first, and then all the structures will be copied back to their original medium(s).

If you specify `SYNTAX`, the generate parameters will be syntactically checked and DMSREO will produce a reorganization summary for your review, but no reorganize actions will actually be performed. Then, you can use the `RESTART` option at a later time to initiate the previously specified reorganize actions.

## File Format Conversion

Another useful feature of reorganization is the ability to change the physical attributes of a structure. During the life of a database, changes such as `BLOCKSIZE` or `LOADFACTOR` can greatly improve the efficient use of the database.

To perform file format conversion, which is similar to the procedure for garbage collection, proceed as follows:

1. Before converting the files, it is recommended that you:
  - a. Dump the database files to a backup medium.
  - b. Dump the audit files to a separate backup medium.
  - c. Delete audit files on disk or diskpack.
2. Insert `$REORGANIZE` at the beginning of the original DASDL source input.
3. Make any corrections to the physical attributes of the structures. You can mark the changes with generate statements, although the reorganization process does not require this.
4. Compile this new source.
5. Initiate the reorganization process with the `IR REORGANIZE` command, `EX DMSDLR`, the `DMSDLR` option of the `DMSUTL` utility, or the `$SET AUTOREORG` compiler control record.
6. It is strongly recommended that you make another backup of the database when the Reorg DBP completes the reorganization.





B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Update, Reorganization, and Purging

---

Example 1: Change data set physical options and set physical options

Original DASDL Source:

```
D DATA SET
(
  A ALPHA (10);
  B BOOLEAN;
  N NUMBER (S5,2);
  R NUMBER (12);
);

S SET OF D
  KEY A;

D DATA SET
(
  AREAS = 10,
  AREALENGTH = 100 BLOCKS,
  BLOCKSIZE = 5 RECORDS,
  CHECKSUM RESET
);
```



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Update, Reorganization, and Purging

---

Step 1 of reorganization: Change DASDL source, as follows, and recompile.

```
?COMPILE TESTDB DASDL LIB
?DATA CARD
$SET REORGANIZE           %added
D DATA SET

(
  A ALPHA (10);
  B BOOLEAN;
  N NUMBER (S5,2);
  R NUMBER (12);
)

S SET OF D
  KEY A;

D
(
  AREAS = 10,
  AREALENGTH = 50 BLOCKS,      %changed
  BLOCKSIZE = 20 RECORDS,     %changed
  CHECKSUM SET                 %changed
);
S                               %added
(                               %added
  AREAS = 20,                  %added
  AREALENGTH = 50 BLOCKS,      %added
  BLOCKSIZE = 200 ENTRIES,     %added
  LOADFACTOR = 90,             %added
  CHECKSUM SET                 %added
);                             %added
GENERATE D,S;                  %added - optional
```

Step 2: Start Reorg DBP with the DMSUTL utility, EX DMSDLR, or the IR command.



Example 2: Changing MODULUS of an Access

Original DASDL source:

```
R RANDOM DATA SET
(
  F NUMBER (10);
  G GROUP
  (
    M NUMBER (6);
    N NUMBER (S5,2);
  );
);
A ACCESS TO R
  KEY IS G
  DUPLICATES
  MODULUS = 500;
```

Step 1 in reorganization: Change DASDL source, as follows, and recompile.

```
?COMPILE TESTDB DASDL LIB.
?DATA CARD
$SET REORGANIZE                                %added
R RANDOM DATA SET
(
  F NUMBER (10);
  G GROUP
  (
    M NUMBER (6);
    N NUMBER (S5,2);
  );
);
A ACCESS TO R
  KEY IS G
  DUPLICATES
  MODULUS = 1000;                                %changed
  GENERATE R;                                    %added - optional
```

Step 2: Start Reorg DBP with the DMSUTL utility, EX DMSDLR, or the IR command.



## Record Format Conversion

During the reorganization process, data items within a data set may change in size, type, offset, and number of occurrences; these changes are subject to the following restrictions.

The DASDL compiler allows data items to be added to or deleted from the description of a data set. When you delete a data item, the DASDL compiler removes the data associated with that item from all records in the data set. When you add a data item, the DBP inserts a data field containing an INITIALVALUE which defaults to nulls (hexadecimal Fs) into all records in the data set. Items that you add can be REQUIRED and can be used as keys of sets or subsets. If an added item is to be REQUIRED or used as a key, you must give it an INITIALVALUE. If you add a key item or a REQUIRED item without an INITIALVALUE, the result will be an error during the DASDL compile.

For a set or subset with a large population, use the following procedure to add a data item that will be a key of the set or subset:

1. Add the data item and remove the set from the DASDL source. Do a \$REORGANIZE compile and run the Reorg DBP. This adds the new item with INITIALVALUES.
2. Create and run a user program that puts into the database the real values of the new item.
3. Add the set or subset back into the DASDL source. Do a \$REORGANIZE compile and run the Reorg DBP. This creates the set ordered by the appropriate values.

If you do not use steps 1-3, the DBP will do the following things:

- The new data item will be initialized to the INITIALVALUE.
- The set or subset will be reordered according to the keys that have INITIALVALUES.
- The user program will change the values of the new items.
- The set or subset will have to be reordered again according to the real values.

The procedure in steps 1-3 usually involves less time and fewer I/Os (Inputs/Outputs).

You may change data item sizes. If the old size is greater than the new size, the DBP will truncate the data from the item. DASDL detects the truncation and gives a warning message. If the new size is larger than the old size, the rules of Table 3-1 apply to zero and blank fill. Null items remain null.

You may add or delete the sign field of a data item. DASDL will detect deletion of the sign field and gives a warning message. If you delete the sign, the Reorg DBP will truncate the sign from the item as the item is moved. If you add the sign, the Reorg DBP will generate a positive sign as the item is moved.

You may change item types. Table 3-1 shows all possible type changes and the effects of each on the original data. Also, Table 3-1 shows the effect of an item size change when type remains unchanged.

The rules for item conversion were designed to adhere as closely as possible to the conventions used for the MOVE statement in COBOL ANSI-74. In line with these conventions, DASDL treats ALPHA items as alphanumeric and BOOLEAN items as 1-digit numeric items of which an even value is FALSE and odd value is TRUE. Where group items are involved in transformations which are illegal in COBOL ANSI-74, DASDL treats them as ALPHA items and allows the conversion.





**Table 3-1. Record Format Conversion**

| Reorganization Move |        | Truncation Or:         |                      |                       | Translation |
|---------------------|--------|------------------------|----------------------|-----------------------|-------------|
| From                | To     | Space Fill<br>on Right | Zero Fill<br>on Left | Zero Fill<br>on Right |             |
| Alpha               | Alpha  | x                      |                      |                       |             |
| Alpha               | Group  | x                      |                      |                       |             |
| Alpha               | Number |                        | x                    |                       | x           |
| Group               | Alpha  | x                      |                      |                       |             |
| Group               | Group  | x                      |                      |                       |             |
| Group               | Number |                        |                      | x                     | x           |
| Number              | Alpha  | x                      |                      |                       | x           |
| Number              | Group  | x                      |                      |                       | x           |
| Number              | Number |                        | x                    |                       |             |

If you specify an INITIALVALUE for an item, all occurrences of the item in the reorganized data set are at first given the INITIALVALUE. Then the DBP moves the values from the "old" data set into the reorganized data set. For an item that you are adding with no INITIALVALUE, the DBP sets the item to NULL (all hexadecimal Fs) before the values are moved in from the "old" data set.

Booleans may be added to or deleted from the end of an existing field item. Do this by specifying the field with the desired modifications in the DASDL reorganization run. Existing Booleans in the field which are included in the new specification for the field retain their information across the reorganization.

If you change items in a data set, you must also change remap invocations of those items. Item changes also force the recompilation of all user programs that invoke the data set or remap. The DBP will invalidate a remap, if it invokes items that you have changed in the data set.

The number of nesting levels in a data item can change, as can the number of occurrences. If you change the number of occurrences without changing the number of levels, then each level is preserved and each subscript remains unchanged. If the number of occurrences decreases in the new database, the DBP will move only the first <n> occurrences to the new record, where <n> represents the number of occurrences in the new record. DASDL detects this condition and gives a warning message. If you increase the number of occurrences in the new database, the DBP moves all the occurrences from the old data set and fills the remaining occurrences with NULL values.

Figure 3-5 shows a record format conversion that transforms PERSON from DATABASE A to DATABASE B. Table 3-2 shows how the transformation takes place. Note that there are three slots in database B without valid data. Going from B to A, the DBP would lose three occurrences of PERSON and give a warning message.

If the number of levels changes (see Figure 3-5), then the DBP recreates the structure linearly and the old subscripts may have no relation to the new.



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Update, Reorganization, and Purging

---

DATABASE A

```
JOB GROUP OCCURS 3 TIMES;
(
    .
    .
    .
    PERSON ALPHA (10) OCCURS 2 TIMES;
)
    .
    .
    .
```

DATABASE B

```
JOB GROUP OCCURS 3 TIMES;
(
    .
    .
    .
    PERSON ALPHA (10) OCCURS 3 TIMES;
)
    .
    .
    .
```

DATABASE C

```
JOB GROUP OCCURS 2 TIMES;
(
    .
    .
    .
    JOB-ORDER OCCURS 2 TIMES;
    (
        .
        .
        .
        PERSON ALPHA (10) OCCURS 2 TIMES;
    )
    .
    .
    .
);
```

Figure 3-5. Sample Databases During Record Format Conversion



**Table 3-2. Mappings of PERSON Between Databases in Figure 3-5**

| DATABASE |       | DATABASE |         |
|----------|-------|----------|---------|
| A        | B     | A        | C       |
| (1,1)    | (1,1) | (1,1)    | (1,1,1) |
| (1,2)    | (1,2) | (1,2)    | (1,1,2) |
| -        | (1,3) | (2,1)    | (1,2,1) |
| (2,1)    | (2,1) | (2,2)    | (1,2,2) |
| (2,2)    | (2,2) | (3,1)    | (2,1,1) |
| -        | (2,3) | (3,2)    | (2,1,2) |
| (3,1)    | (3,1) | -        | (2,2,1) |
| (3,2)    | (3,2) | -        | (2,2,2) |
| -        | (3,3) |          |         |

You may change the grouping and levels of data items, subject to the following restrictions:

- Regrouping of items must not cause data to be duplicated. For example, the following transformation is forbidden.

| Old         | New         |
|-------------|-------------|
| A GROUP     | A ALPHA(2); |
| B ALPHA(1); | B ALPHA(1); |
| C ALPHA(1); | C ALPHA(1); |

In this example, a duplication of the data represented by A occurs in the new definition, because B and C both contain data that is present in the new A.

- Regrouping of items may not cause multiple mapping of information into an item. This would occur if the new definition were transformed into the old definition in the preceding example.

## Version Checking

Each structure and remap has a version stamp associated with it. The version stamp reflects the time of the latest change in the logical description of the database that would affect any programs that use that structure or remap. Any program with a different version stamp will cause a version error when you open the database. The DBP requires a recompilation of the program to bring it up to date with the current description of that structure.

### NOTE

Whenever you delete a structure or remap, any attempt to use that structure or remap results in a version error.

Some of the changes that are legal with reorganization require that the version stamps of some of the structures or remaps change. You should be aware of any changes requiring recompilation and the magnitude of the recompilation effort required before making any changes to the database.



## Record Format Change Pragmatics

The following paragraphs summarize the changes that you can make to record formats.

### Data Item Changes

DASDL allows the following changes to data items in record format, with these exceptions:

- You may not change any item you use as a key of a set, subset, or access.
- You may not change FIELDS except to add new BOOLEANS at the end.
- You may not change BOOLEANS.
- You may not change a BOOLEAN in a FIELD.

#### ALPHA to NUMBER

The Reorg DBP zone-strips and right-justifies the data either by truncation or by using leading zeros. If the NUMBER is signed, the Reorg DBP gives it a positive sign. If the NUMBER is scaled, the Reorg DBP assumes that the ALPHA item has the same implied scale factor. Trailing blanks will become significant numbers in the result.

#### NUMBER to ALPHA

Any sign or scale factor is lost. The Reorg DBP justifies the data to the left by truncation or with trailing blanks, and fills the zone with hexadecimal Fs.

#### GROUP to ALPHA

The Reorg DBP treats the GROUP as if it were a single ALPHA item (that is, the Reorg DBP will not do truncation on fields within the GROUP). Items within the GROUP are lost as separate entities and new items with the same names may not be added at the same time.

#### GROUP to NUMBER

The Reorg DBP treats the GROUP as if it were a single ALPHA item, even if there are NUMBER items within the GROUP.

#### ALPHA to GROUP

The Reorg DBP does not do mapping to individual items within the GROUP because it treats the GROUP as a single ALPHA item.

### ADD/DELETE Items

DASDL allows record format changes to the following ADD/DELETE items.

ALPHA  
NUMBER  
BOOLEAN (if not in a FIELD)  
GROUP  
FIELD  
BOOLEAN in a FIELD (if it is the last item in the FIELD)





The following paragraphs describe the changes DASDL allows in displacement, size, and sign.

#### Displacement

Displacement refers to changing the order of data items. DASDL allows displacement for all items and for BOOLEANS in a FIELD.

#### Size

Where DASDL allows it, you may increase or decrease the size of items by truncation or padding, as long as you do this according to the conventions of COBOL ANSI-74. Because a BOOLEAN is a fixed size, the DBP does not allow size change.

#### Sign

You may add a sign to or delete it from any NUMBER item. When you add it, the sign is positive. You cannot change the sign, when the NUMBER item is a key.

### Item Options

The following item options may be affected by record format conversion.

#### REQUIRED Option

You can add the REQUIRED option to an existing item. You can add a new item with the REQUIRED option set only if you specify an INITIALVALUE. You can delete REQUIRED on an update or reorganization run.

#### OCCURS Option

You can add the OCCURS option to or delete it from an item or increase or decrease its value. You can move an item with or without the OCCURS option into a GROUP with or without the OCCURS option. You cannot add OCCURS to an item that is a key.

#### INITIALVALUE Option

You can add or change INITIALVALUE without affecting the contents of existing records.

### Structure Changes

You can make the following changes with record format conversion.

#### VERIFY Clause

You can add or change the VERIFY clause. You can delete the VERIFY clause with an update run. The Reorg DBP will check all existing records.

#### WHERE Clause

You can change or add the WHERE clause for disjoint sets or subsets only.

#### REQUIRED ALL Option

You can add the REQUIRED ALL option as long as you do not also add new items. The Reorg DBP will check all fields for non-null value.



### Embedded Data Sets

You can add or delete embedded data sets as long as you generate the parent data set. When you add, you must also add the set if it is a standard data set. When you delete, you must also delete associated sets and data sets. You cannot use a structure change to change the level of embeddedness.

#### NOTE

You can add or delete disjoint data sets with an update run.

### Embedded Sets and Subsets

You can add or delete embedded sets and subsets as long as you generate the parent data set. This applies to embedded automatic sets and subsets (if you also add the data set) and embedded manual subsets (the data set can be existing or new). You can delete manual and automatic subsets (as well as embedded automatic sets) as long as at least one set remains for each embedded standard data set. You cannot add an embedded automatic set or subset to an existing data set.

#### NOTE

You can add disjoint sets and subsets.

## Performing Record Format Conversion

Proceed as follows to perform record format conversion.

1. Recompile the modified database description with DASDL. You must include the entire database DASDL source. Tables 3-4 through 3-9 indicate which changes DASDL allows you to make to the database during record format conversion.

You must place the \$REORGANIZE statement in front of the database description. This statement instructs DASDL to update the existing database description.

A GENERATE statement can appear for each structure being reorganized.

2. Run the Reorg DBP. The DASDL compiler names the Reorg DBP XXXREO where XXX represents the first three characters of the database name. Use the IR command, EX DMSDLR, the DMSDLR option of the DMSUTL utility, or the \$SET AUTOREORG compiler control record to execute the reorganization process. The Reorg DBP reads the old data set, set, or subset and generates a new one in the new format. The Reorg DBP continues this process until it generates all data sets, sets, and subsets.
3. You must recompile all programs that invoke the reorganized structure directly. When you delete items, you must also remove all references to these items from user programs.

You do not need to recompile programs that invoke remaps of the reorganized structure as long as the remap record format remains the same.



### Example 1: Data Set Record Format Changes

Original DASDL source:

```
PARAMETERS
(
  ALLOWEDCORE = 1000000
);

D DATA SET
(
  A ALPHA (10);
  B BOOLEAN;
  N NUMBER (S5,2);
);

S SET OF D
  KEY A;
```

Step 1 of reorganization: Change DASDL source, as follows, and recompile.

```
$ REORGANIZE                                %added
PARAMETERS
(
  ALLOWEDCORE = 1200000                      %changed
);

D DATA SET
(
  A ALPHA (10);
  G GROUP                                    %added
  (                                          %added
    P ALPHA (8);                          %added
    Q NUMBER (12);                        %added
  );                                       %added
  N NUMBER (S5,2) INITIALVALUE 0; %changed
);

S SET OF D
  KEY A
  GENERATE D;                               %added - optional
```

Step 2: Start Reorg DBP with the DMSUTL utility, EX DMSDLR, or the IR command.

In Example 1, note that the ALLOWEDCORE parameter was changed. This is possible because all changes permitted for the update function are also permitted for the reorganization function. Also note that Item B was deleted.



Example 2: Data Set Record Format Changes

Original DASDL source:

```
K DATA SET
(
  B BOOLEAN;
  N NUMBER (3);
  R NUMBER (12);
  A ALPHA (199);
  O NUMBER (6) OCCURS 100 TIMES;
  F ALPHA (16) ;
  G GROUP
  (
    G1 ALPHA (29);
    G2 NUMBER (8);
  );
);

R-K REMAPS K
(
  B;
  N;
  R;
  A;
  O;
  F;
  G;
);
LDB DATABASE (R-K);
```

Step 1 in reorganization: Change DASDL source, as follows, and recompile.

```
$ REORGANIZE                                %added
K DATA SET
(
  R NUMBER (12);                             %lines deleted here
);

R-K REMAPS K                                %lines deleted here
(
);

LDB DATABASE (R-K);
GENERATE K;                                %added - optional
```

Step 2: Start Reorg DBP with the DMSUTL utility, EX DMSDLR, or the IR command.





Example 2: Record Format Changes to Delete Embedded Structure

Original DASDL source:

```
D DATA SET
(
  E DATA SET
  (
    E1 ALPHA (10);
    E2 NUMBER (12);
  );
  S-E SET OF E
    KEY E1;
  R-E REMAPS E
  (
    E1;
    E2;
  );
  D1 BOOLEAN;
  D2 ALPHA (5);
);

S-D SET OF D
  KEY D2;

R-D REMAPS D
(
  P;
  R-E;
  S-E;
  D1;
  D2;
);

LDB DATABASE (R-D);
```



Step 1 in reorganization: Change DASDL source, as follows, and recompile.

```
$ REORGANIZE                                %added
D DATA SET
(
  D1 BOOLEAN;                                %lines deleted here
  D2 ALPHA (5);
);

S-D SET OF D
  KEY D2;

R-D REMAPS D
(
  D1;
  D2;
);

LDB1 DATABASE (R-D);
```

Step 2: Start Reorg DBP with the DMSUTL utility, EX DMSDLR, or the IR command.

## RESTART

Use the RESTART function to restart a reorganize run that did not complete because of a nonerror failure. You may also use this function to initiate a reorganize run that you had entered previously as a SYNTAX run.

You can initiate a RESTART function with the IR RESTART command, with EX DMSDLR, or with the DMSDLR restart option in the DMSUTL utility. The following description assumes that you are using the IR command syntax, but the DMSUTL screen choices are similar.

When you initiate a RESTART function, the IR syntax that initiated the failed reorganize run will be displayed on the ODT along with some instructions and a list of action choices. You may have to reload some of the structures from the dump that you made before you initiated the reorganization (they will be specified in the instructions that are displayed), and then you must enter one of the following with an ACCEPT command (for example, 12 AX RES where 12 is DMSDLR's mix number):

- RES - to restart the failed reorganization or to initiate a reorganization that you entered previously as a SYNTAX run.
- REM - to remove the database's memory of the failed reorganization and go to EOJ. Note that this choice will probably leave the database in an inconsistent state because the partial results of the failed reorganization are not backed out. You should then reload all of the files from the dump you made before you initiated the reorganization.
- EXI - to exit from the restart function without doing anything.
- SYN - to perform another SYNTAX run (and create another reorganization summary). This choice is only offered if you did a SYNTAX run originally.
- RGO - to do the same thing as REM.



A reorganize run proceeds through several phases during its execution. Certain information in the control record of the Reorganization Information File (XXXRNF) is used to keep track of the current phase during the reorganization process. During a restart, this information is checked and the failed reorganization will be restarted in or before the phase it was in when the failure occurred.

If the reorganization failed in the initialization phase, it will be restarted from the very beginning.

If it failed in the file conversion phase, it usually will be restarted within that phase at the beginning of the current reorganization action. Depending on the reorganization action and the type of structures, you may have to reload some of the structures from the dump you made before initiating the reorganization.

If the reorganization action at the time of the failure was data set compression (garbage collection), the reorganization will be restarted at the beginning of compression on the current data set. If the current data set is disjoint, you will not be required to reload any files. Because embedded unordered data set compression fixes up the master data set while it does the compression, you will be instructed to reload the master data set in that situation.

If the reorganization action at the time of the failure was fix-up, you will be instructed to reload the current structure file. Then, the reorganization will be restarted at the beginning of the fix-up of that structure.

During the compression of a data set, a work file is created (XXX00W) that contains a mapping of the old Relative Record Numbers (RRNs) to the corresponding new ones. This file should be present when a fix-up of one of the associated sets is restarted. If not, you will be instructed instead to reload the current data set and all its referencing sets and subsets that the fixup affected. The reorganization will then be started at the beginning of compression on the current data set.

The generation of all the new sets and subsets that will reference a specific data set is done simultaneously. This type of generation is called a generate-new. If the reorganization action at the time of the failure was generate-new, the reorganization will be restarted at the beginning of the generate-new for the current referenced data set, and you will not be required to reload any files.

If the reorganization action at the time of the failure was set balance, the reorganization will be restarted at the beginning of the balance of the current set or subset. If the set or subset was disjoint, you will not be required to reload any files. Because the fixup of the master data set is done at the same time as the set balancing of its embedded set or subset, you will be instructed to reload the master data set in that situation.

If the reorganization failed in the copy phase, it will be restarted at the beginning of the copying of the current structure file.

If the reorganization failed in the finish-up phase, it will be restarted at the beginning of that phase.

If you initiate a reorganization run and DMSDLR detects that a previous reorganization has failed, it will first perform a reorganize restart. You will be offered the same choices as for restart reorganize, except for the following:

- RG0 - to remove the database's memory of the failed reorganization and proceed with the newly entered reorganization.

For more information about the IR RESTART command, see the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 2: Database Administration*.

For more information about the RESTART option in DMSUTL, see the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*.



## PURGING

In some cases, the database administrator (DBA) may need to empty certain database structures of the data they contain, to prepare these structures to receive new data. This may have to be done regularly - at the end of each month, for example. Because it is time consuming to clear an entire structure, record by record, DMSII provides the PURGE function so that the DBA can perform this function quickly and simply. When a structure is purged, the data blocks are deleted from the structure, leaving an empty structure that takes up only a small amount of space.

### NOTE

A purge cannot be done when user programs are accessing the database.

PURGE can be executed with the IR command or with the DMSDLR function of the DMSUTL utility.

If the operating system fails during a purge, recovery can be done with either a halt/load, an IR RE-BUILD command, or with the DMSDLR function of the DMSUTL utility.

The following rules apply to the PURGE function:

- If a data set is purged, all structures embedded in it are automatically purged.
- Data sets and manual subsets, whether disjoint or embedded, can be explicitly purged.
- If a data set is purged, all associated sets and subsets are automatically purged. These associated sets can be disjoint, embedded in the purged data set, or embedded in another data set.
- Sets and automatic subsets cannot be explicitly purged.
- Once you use a PURGE function on a structure, it cannot be returned to the state it was in before the PURGE. This is because audit trail records are not kept of each change made during the PURGE. Therefore, a rollback recovery will be stopped at the audit trail discontinuity caused by a PURGE.

## RELEASE UPDATES

Table 3-3 shows how to convert databases for use with new releases of the DMSII software. In general, recompile the DASDL source with the new release of the DASDL compiler and set the \$UPDATE compiler option. Normally, user programs do not have to be recompiled unless you want to use features of the host language compiler that are introduced in the new release.





Table 3-3. Converting Databases for New Software Release

| Converting from: | Converting to:     | Action Required:                                                                                          |
|------------------|--------------------|-----------------------------------------------------------------------------------------------------------|
| ASR 7.1          | ASR 7.2            | Database:<br>Recompile DASDL source with \$UPDATE option set.                                             |
| ASR 7.0          | ASR 7.1<br>ASR 7.2 | Database:<br>Recompile DASDL source with \$UPDATE option set.<br><br>User Program:<br>No change required. |
| ASR 6.8          | ASR 7.1<br>ASR 7.2 | Database:<br>Recompile DASDL source with \$UPDATE option set.<br><br>User Program:<br>No change required. |
| ASR 6.7          | ASR 7.1<br>ASR 7.2 | Database:<br>Recompile DASDL source with \$UPDATE option set.<br><br>User Program:<br>No change required. |
| Pre-6.7          | ASR 7.1<br>ASR 7.2 | First convert to ASR 6.7.<br>Then follow instructions to convert to ASR 7.1 or 7.2.                       |



## CHANGES DASDL PERMITS

To change the features of the database, use either the update or the reorganization function. Tables 3-4 through 3-9 indicate which changes the DBP allows and which forms of the update and reorganization functions to use to make each change. Often, more than one form of the update and reorganization functions is appropriate to make the desired change. In this case, select whichever form is most convenient.

Table 3-4. Allowed Changes to Options

| Option     | Update | Reorg |
|------------|--------|-------|
| AUDIT      | No     | No    |
| KEYCOMPARE | Yes    | Yes   |
| STATISTICS | Yes    | Yes   |

Table 3-5. Allowed Changes to Parameters

| Parameter    | Update | Reorg |
|--------------|--------|-------|
| ALLOWEDCORE  | Yes    | Yes   |
| CONTROLPOINT | Yes    | Yes   |
| STACKSIZE    | Yes    | Yes   |
| SYNCPOINT    | Yes    | Yes   |

Table 3-6. Allowed Changes to Audit Trail

|                                   | Update | Reorg |
|-----------------------------------|--------|-------|
| AREAS                             | Yes    | Yes   |
| AREALENGTH                        | Yes    | Yes   |
| BLOCKSIZE                         | Yes    | Yes   |
| CHECKSUM                          | Yes    | Yes   |
| FAMILYNAME                        | Yes    | Yes   |
| UPDATE-EOF                        | Yes    | Yes   |
| DUPLICATED                        | Yes    | Yes   |
| FAMILYNAME of<br>DUPLICATED AUDIT | Yes    | Yes   |

Changes to any audit trail attributes (except UPDATE-EOF and CHECKSUM) affect recovery. If you change FAMILYNAME, it is your responsibility to copy any audit trail files to the correct diskpack, if the DBP needs them for recovery.

Any changes to the other attributes (all attributes except DUPLICATED, CHECKSUM, and UPDATE-EOF) introduce a discontinuity in the audit trail that the recovery process cannot cross. The integrity of the database must not be questioned if you change any audit trail attributes; it is not possible to recover a database through the point of discontinuity.



With DASDL update or reorganization, you can turn **DUPLICATED** on or off and change the **FAMILYNAME** of the duplicated audit trail. These changes affect any recovery process that uses the existing audit trail.

If you add **DUPLICATED AUDIT**, and a recovery process that uses audit trails produced before the change is necessary, you can use the **COPY** option of the **DMSAUD** utility program to produce a secondary audit from the existing primary audit. When you eliminate **DUPLICATED AUDIT**, the recovery process will not use the recovery audit trail if the primary audit trail fails.

**NOTE**

**DUPLICATED AUDIT** adds little overhead to the DBP. The I/O for the duplicated audit is "fired off" at the same time as the primary audit. Waiting is done only for the primary audit.



**Table 3-7. Allowed Changes to Data Sets**

|                                               | Update             | Reorganize      |
|-----------------------------------------------|--------------------|-----------------|
| <b>Data sets</b>                              |                    |                 |
| Add new disjoint data set                     | Yes                | Yes             |
| Add new embedded data set                     | Yes (1)            | Yes (1)         |
| Delete existing disjoint data set             | Yes (2)            | Yes (2)         |
| Delete existing embedded data set             | Yes (2),<br>(4)    | Yes (2),<br>(4) |
| Change data set name                          | No                 | No              |
| Change data set type                          | No                 | No              |
| Add, change, or<br>delete verify condition    | Delete<br>only (5) | Yes             |
| Change order in which items are<br>declared   | No                 | Yes             |
| Arrange the data set in the order<br>of a set | No                 | Yes (9)         |
| <b>Data Items</b>                             |                    |                 |
| Add new data item                             | No                 | Yes (7)         |
| Delete existing data item                     | No                 | Yes             |
| Change data item type                         | No                 | Yes             |
| Change data item size                         | No                 | Yes             |
| Change INITIALVALUE option                    | Yes                | Yes             |
| Change OCCURS option                          | No                 | Yes             |
| Change REQUIRED option                        | Delete<br>only (6) | Yes (6)         |
| Change number of occurrences                  | No                 | Yes             |
| Change precision                              | No                 | Yes             |
| Change scale factor                           | No                 | Yes             |
| Change signed option                          | No                 | Yes             |





**Table 3-7. Allowed Changes to Data Sets (Cont.)**

|                                                                    | Update          | Reorganize |
|--------------------------------------------------------------------|-----------------|------------|
| <b>Data items (cont.)</b>                                          |                 |            |
| Add or delete Booleans in an existing field item                   | No              | Yes (8)    |
| Change order of Booleans in an existing field item                 | No              | No         |
| <b>Group Items</b>                                                 |                 |            |
| Add new group item                                                 | No              | Yes        |
| Delete existing group item                                         | No              | Yes        |
| Add or delete items in an existing group item                      | No              | Yes        |
| Change order in which items are declared in an existing group item | No              | Yes        |
| Change OCCURS option                                               | No              | Yes        |
| Change REQUIRED option                                             | Delete only (6) | Yes (7)    |
| Change number of occurrences                                       | No              | Yes        |
| <b>Data Set Physical Options</b>                                   |                 |            |
| AREAS                                                              | No              | Yes        |
| AREALENGTH                                                         | No              | Yes        |
| BLOCKSIZE                                                          | No              | Yes        |
| BUFFERS                                                            | Yes             | Yes        |
| CHECKSUM                                                           | Yes             | Yes        |
| FAMILYNAME                                                         | Yes (3)         | Yes        |
| LOCK TO MODIFY DETAILS                                             | Yes             | Yes        |
| MAXRECORDS                                                         | No              | Yes        |

#### NOTES

1. You can add an embedded data set provided that you also add the master data set.
2. Any program that attempts to use a deleted data set will receive an OPENERROR. You can never delete the restart data set. Deletion of a disjoint data set requires that you delete all sets and subsets of the data set. This includes manual subsets in other data set families.



3. When you change the FAMILYNAME option, the DBP will expect to find the data set on the new device. You can copy the data set from the old device to the new device with the SYSTEM/COPY utility.
4. You can delete embedded data sets provided you delete the master data set at the same time.
5. You cannot specify a new VERIFY statement and you cannot change an existing VERIFY statement. To delete an existing VERIFY statement, use either the UPDATE or REORGANIZE option. The date-timestamps will change for that structure. As a result, you will need to recompile all user programs that invoke the structure.
6. You can add a REQUIRED option only if you give it an INITIALVALUE. If you delete a REQUIRED item, the structure date-timestamp will change and you must recompile all user programs invoking the structure.
7. An item that you are adding can be REQUIRED only if you specify an INITIALVALUE.
8. You can add or delete BOOLEAN items in a FIELD, only if they are the last items in the FIELD.
9. Only standard data set or disjoint unordered data set may use ORDERED BY option.



**Table 3-8. Allowed Changes to Sets, Subsets, and Accesses**

|                                                       | Update  | Reorganize |
|-------------------------------------------------------|---------|------------|
| <b>Sets</b>                                           |         |            |
| Add new disjoint set for existing data set            | No      | Yes (1)    |
| Add new disjoint set(s) for new data set              | Yes     | Yes        |
| Add new embedded set for existing data set            | No      | Yes (8)    |
| Add new embedded set for new data set                 | Yes (7) | Yes (3)    |
| Delete existing disjoint set                          | Yes     | Yes        |
| Delete existing embedded set                          | Yes (5) | Yes (3)    |
| Change set to subset or vice versa                    | No      | Yes (9)    |
| Change set organization                               | No      | No         |
| Change data set referenced by set                     | No      | No         |
| Change set keys                                       | No      | Yes (9,10) |
| Add or delete set keys                                | No      | Yes (9,10) |
| Change size of key items                              | No      | Yes (9)    |
| Add/change/delete key data                            | No      | Yes (9)    |
| Change ASCENDING/DESCENDING option                    | No      | Yes        |
| Change DUPLICATES option                              | Yes (2) | Yes (2)    |
| Generate existing disjoint set from existing data set | No      | Yes        |



**Table 3-8. Allowed Changes to Sets, Subsets, and Accesses (Cont.)**

|                                                                            | Update  | Reorganize |
|----------------------------------------------------------------------------|---------|------------|
| <b>Subsets</b>                                                             |         |            |
| Add new disjoint<br>automatic or manual<br>subset for existing<br>data set | Yes (1) | Yes (1)    |
| Add new disjoint automatic<br>or manual subset<br>for new data set         | Yes     | Yes        |
| Add new embedded subset                                                    | Yes (7) | Yes (7)    |
| Delete existing disjoint<br>automatic or manual<br>subset                  | Yes     | Yes        |
| Delete existing embedded<br>automatic or manual<br>subset                  | Yes (5) | Yes (5)    |
| Change automatic subset to<br>manual subset or vice versa                  | No      | No         |
| Change subset to set or<br>vice versa                                      | No      | Yes (9)    |
| Change subset organization                                                 | No      | No         |
| Change data set referenced<br>by subset                                    | No      | No         |
| Change subset keys                                                         | No      | Yes (9)    |
| Add or delete subset keys                                                  | No      | Yes (9,10) |
| Change size of key items                                                   | No      | Yes (9)    |
| Add/change/delete key data                                                 | No      | Yes (9)    |
| Change or add ASCENDING/DESCENDING<br>option                               | No      | Yes (9)    |
| Change DUPLICATES option                                                   | Yes (2) | Yes (2)    |
| Change WHERE condition                                                     | No      | Yes (9)    |
| <b>Accesses</b>                                                            |         |            |
| Add disjoint access for new<br>data set                                    | Yes     | Yes        |
| Delete existing access                                                     | Yes (4) | Yes (4)    |
| Change data set referenced<br>by access                                    | No      | No         |
| Change access key item(s)                                                  | No      | No         |
| Change DUPLICATES option                                                   | Yes (2) | Yes (2)    |
| Change MODULUS                                                             | No      | Yes        |





Table 3-8. Allowed Changes to Sets, Subsets, and Accesses (Cont.)

| Set/Subset Physical Options | Update  | Reorganize |
|-----------------------------|---------|------------|
| AREAS                       | No      | Yes        |
| AREALENGTH                  | No      | Yes        |
| BLOCKSIZE                   | No      | Yes        |
| BUFFERS                     | Yes     | Yes        |
| CHECKSUM                    | Yes     | Yes        |
| FAMILYNAME                  | Yes (6) | Yes        |
| LOADFACTOR                  | Yes     | Yes        |
| MAXENTRIES                  | No      | Yes        |

#### NOTES

1. Adding a new set or subset necessitates that all key items contain valid data. That is, the value contained in the key items can not be NULL (all bits on).

Declaring a new set does not automatically insert applicable records for existing data sets. You must use reorganization to generate the set from the data set. You must ensure that all existing data set records have valid key values. Key items must not be NULL and, if the set or subset does not permit duplicates, all key values must be unique. If reorganization encounters a record with a NULL or duplicate key, it terminates with an exception.

It is impossible for the reorganization process to determine the sequence in which records were originally added to the data set. Therefore, when a set or subset specifying DUPLICATES FIRST or DUPLICATES LAST is generated from the data set, existing records are not stored in chronological order. The DBP maintains all records you add after reorganization in proper sequence.

When you delete a set or subset, any program that attempts to use it will receive an OPENERROR.

2. You can change NO DUPLICATES to DUPLICATES, DUPLICATES FIRST, or DUPLICATES LAST. DASDL does not permit any other changes.
3. You can add a new embedded access only if both the data set containing the access and the data set referenced by the access are new.
4. You can delete existing accesses only if you also delete the data set they reference.
5. You can delete embedded sets and subsets if you delete the master at the same time, or if at least one embedded set remains for the master data set.
6. When you change the disk/pack option, the DBP will expect to find the set or subset on the new device. You can copy the set or subset from the old device to the new device with the SYSTEM/COPY or the DMPALL utility.



7. You can add new embedded sets and subsets as long as you add the master at the same time.
8. You cannot add embedded automatic sets or subsets.
9. These capabilities apply only to disjoint sets and disjoint automatic subsets.
10. You can use an added data item as a key only if you specify an INITIALVALUE.

**Table 3-9. Allowed Changes to Remaps**

|                                                  | Update  | Reorganize |
|--------------------------------------------------|---------|------------|
| <b>Remaps</b>                                    |         |            |
| Add new disjoint remap                           | Yes     | Yes        |
| Add new embedded remap<br>to data set            | Yes     | Yes        |
| Delete existing disjoint remap                   | Yes     | Yes        |
| Delete existing embedded remap                   | Yes     | Yes        |
| Change remap name                                | No      | No         |
| Change data set referenced<br>by remap           | No      | No         |
| Change order in which data<br>items are declared | Yes (1) | Yes (1)    |
| <b>Remap Options</b>                             |         |            |
| Change READONLY all                              | Yes     | Yes        |
| Change REQUIRED all                              | Yes     | Yes        |
| <b>Remap Data Items</b>                          |         |            |
| Add new remap data item                          | Yes (1) | Yes (1)    |
| Delete existing remap data item                  | Yes (1) | Yes (1)    |
| Change remap data item name                      | No      | No         |
| Change HIDDEN option                             | Yes (2) | Yes (2)    |
| Change READONLY option                           | Yes     | Yes        |
| Change REQUIRED option                           | Yes     | Yes        |



**Table 3-9. Allowed Changes to Remaps (Cont.)**

|                                                                                                  | Update  | Reorganize |
|--------------------------------------------------------------------------------------------------|---------|------------|
| <b>Remap Group Items</b>                                                                         |         |            |
| Add new remap group item                                                                         | Yes (1) | Yes (1)    |
| Delete existing remap group item                                                                 | Yes (1) | Yes (1)    |
| Change remap group item name                                                                     | No      | No         |
| Change order in which items in<br>remap group item are declared                                  | No      | No         |
| Change HIDDEN option                                                                             | Yes (2) | Yes (2)    |
| Change READONLY option                                                                           | Yes     | Yes        |
| Change REQUIRED option                                                                           | Yes     | Yes        |
| <b>Remap References</b>                                                                          |         |            |
| Add new remap reference                                                                          | Yes     | Yes        |
| Delete existing remap reference                                                                  | Yes (1) | Yes (1)    |
| Change remap reference name                                                                      | No      | No         |
| <b>Remap Set-Subset-Access</b>                                                                   |         |            |
| Add new remap set-subset-access                                                                  | Yes     | Yes        |
| Delete existing remap set-subset-access                                                          | Yes     | Yes        |
| Change remap set-subset-access name                                                              | No      | No         |
| <b>Logical Databases</b>                                                                         |         |            |
| Add new logical database                                                                         | Yes     | Yes        |
| Delete existing logical database                                                                 | Yes     | Yes        |
| Change logical database name                                                                     | No      | No         |
| Change structures included in<br>logical database                                                | Yes     | Yes        |
| Change disjoint sets, subsets, and<br>accesses included in the logical<br>database               | Yes     | Yes        |
| Change the name of data sets,<br>sets, subsets, accesses, and remaps<br>in the logical database. | Yes     | Yes        |

## NOTES

1. DASDL permits changes that alter the format of the remap record. These changes will result in a new date-timestamp for the remap structure. You must recompile all user programs that invoke the changed remap.
2. You cannot delete HIDDEN from a remap data item or a remap group item.



## SECTION 4

### AUDIT AND RECOVERY

#### INTRODUCTION

Because hardware and/or software failures may corrupt the information held in a database, DASDL provides an option for auditing all changes to a particular database. The database program (DBP) records all changes to the database in a series of files (the audit trail). In the event of a hardware or software failure, the DBP uses the information in the audit trail to recover the database.

#### TRANSACTIONS AND SYNCPOINTS

The audit and recovery procedures that the Data Management System II (DMSII) provides are based on the concept of transactions. A transaction is a logical update to a database and consists of a user program executing a **BEGIN-TRANSACTION** statement, followed by one or more data management operations, followed by an **END-TRANSACTION** statement. After executing a **BEGIN-TRANSACTION** statement a user program is said to be in transaction state until it executes an **END-TRANSACTION** statement. The user program must not have nested transactions, and operations that change a database must occur only in transaction state. An **END-TRANSACTION** statement also unlocks any records that the program still has locked (except for the restart data set record).

A partial transaction is a transaction that the system fails to complete for some reason. The DBP must always back out the effects of partial transactions on a database to preserve the integrity of the database. An example of a partial transaction is a new record insertion into a data set without an update of the associated sets. In a multiprogramming environment, the DBP must back out overlapping transactions because the information the DBP has backed out for the partial transaction may have been used by other transactions.

When the DBP has to back out a partial transaction, it has to back out all transactions to a point where no user program was in transaction state. Because this process is inefficient and time consuming, the DBP provides syncpoints. A syncpoint guarantees that no user program is in transaction state.

The database administrator (DBA) may specify, in DASDL, the frequency of syncpoints in terms of transactions (for example, a syncpoint should occur after every *n* transactions, where *n* is some small integer value). Whenever a user program executes an **END-TRANSACTION** statement, the DBP for the corresponding database checks to see if *n* transactions have taken place since the last syncpoint. If *n* transactions have taken place, the DBP:

1. Allows in-process transactions to complete and suspends any new **BEGIN-TRANSACTION** requests.
2. Places a syncpoint record in the current audit block.
3. Initiates writes of the primary and secondary audit buffers to the audit trail.

When the DBP physically completes the writes to the audit blocks, it reinstates all suspended jobs.





Additional syncpoints may also occur for the following reasons:

- When a user program executes an **END-TRANSACTION** and there are no other programs in transaction state. This pseudo-syncpoint causes the DBP to place a syncpoint record in the audit block, but does not initiate a write to the audit block.
- When a user program executes an **END-TRANSACTION** with **SYNC** statement, the DBP does not return control to the user program until after the completion of the syncpoint.
- When a user program executes a database **CLOSE** statement, the DBP does not return control to the user program until after completion of the syncpoint. Such a syncpoint ensures that the DBP will never back out a **CLOSE** statement during the process of backing-out overlapping transactions.

## CONTROLPOINTS AND THE FLUSHING OF DATA BUFFERS

To minimize physical Input/Output (I/O), the DBP accumulates changes to a structure file block in a database buffer in memory and writes them to the appropriate medium when that buffer is required for another purpose. Before the DBP initiates any physical write of a data buffer, the DBP must ensure that all changes to the block have already been physically written to the audit trail; because one of the key principles underlying the audit/recovery process is that any change must be physically recorded in an audit trail before being made to a database. A description of the synchronization of the physical writing of audit buffers and data buffers appears later in this section.

Leaving data buffers to be written out in this haphazard manner is highly undesirable. For recovery purposes, it is helpful to have a sequence of established points at which the DBP flushes data buffers. For this purpose, the DBP provides controlpoints.

The DBA may specify, in DASDL, the frequency of controlpoints in terms of syncpoints (for example, a controlpoint may occur every  $n$  syncpoints, where  $n$  is some small integer value). When the DBP completes a syncpoint, it checks to see if it is also a controlpoint. If it is a controlpoint the DBP does the following:

- Physically writes all modified data buffers that have not been flushed since before the last controlpoint (because a controlpoint is also a syncpoint, the DBP has already flushed the audit buffers).
- Writes a controlpoint record to the audit trail, when the DBP I/Os have been completed.

This approach guarantees that the DBP physically writes every update to a database block to disk or diskpack at least at the second controlpoint after you make the update. Because of this, halt/load recovery only needs to back up two controlpoints before applying after images.

It is possible to physically write out all updated data buffers at every controlpoint. However, this considerably increases the amount of I/O performed without introducing significant compensatory benefits. Note that the DBP simulates two controlpoints whenever a user program physically closes the database by writing a database **CLOSE** record to the audit trail and flushing all updated data buffers.



## AUDIT BUFFER AND DATA BUFFER SYNCHRONIZATION

As mentioned earlier, the DBP must not write any updated structure file block to a database before it writes the audit records containing the changes to the audit trail. The DBP uses two variables in connection with this synchronization:

AWC = Audit Write Count (count of physical writes initiated to the audit trail).

WAN = Written Audit Number (last value of AWC with I/O complete).

Then, the DBP provides synchronization, as follows:

1. The DBP notes the current value of AWC whenever a user program updates a structure file block in a data buffer.
2. Before initiating a physical write of a data buffer, the DBP retrieves the value of AWC corresponding to the last update of the buffer. If that value is greater than WAN, the DBP must physically write audit buffers to the audit trail before it physically writes the data buffer to disk or diskpack.

## AUDIT CHANGE NUMBERS AND AUDIT BLOCK NUMBERS

During a recovery process it is necessary to know whether the DBP has physically carried out certain changes recorded in an audit trail to the relevant blocks in the database structure files. The DBP determines this with 18-digit Audit Change Numbers (ACNs). Each ACN consists of a 12-digit Audit Block Number (ABN) followed by a 6-digit Audit Record Offset (ARO). Every record in an audit trail has a unique ACN.

The control information for every block in every database structure file contains an 18-digit audit change number field. The DBP initializes this field to zero and, before any change to the block is (logically) written out, alters it to the ACN of the audit trail record containing details of the change. The DBP records the previous value of the field in the audit trail record. Therefore, the DBP identifies an audit trail record with an ACN and each audit trail record contains the Previous Audit Change Number (PACN).

The DBP uses the following algorithm to apply before images to database files on the basis of information recorded in the audit trail. Let ACN-AUDIT represent the audit change number of the audit trail record and ACN-STRUCTURE represent the value of the audit change number recorded in the structure file block. The various possibilities are:

- If ACN-AUDIT is less than ACN-STRUCTURE, an error exists.
- If ACN-AUDIT is equal to ACN-STRUCTURE, apply the before image.
- If ACN-AUDIT is greater than ACN-STRUCTURE, ignore the change.



Similarly, the DBP uses the following algorithm to apply after images to database files on the basis of information recorded in the audit trail; where PACN-AUDIT is the value of the previous audit change number recorded in the audit trail record:

- If PACN-AUDIT is less than ACN-STRUCTURE, ignore the change.
- If PACN-AUDIT is equal to ACN-STRUCTURE, apply the after image.
- If PACN-AUDIT is greater than ACN-STRUCTURE, an error exists.

## TYPES OF RECOVERY

The integrity of a database may be compromised for one of four reasons:

- A database user program fails in transaction state. Program abort recovery fixes the corruption this causes; the DBP initiates abort recovery automatically when it is needed.
- A system failure (a halt/load) occurs while a user program is updating the database. Halt/load recovery fixes the corruption this causes. Use the command IR HALTLOAD or the DMSUTL utility to initiate halt/load recovery. When a user program opens the database, the DBP initiates halt/load recovery, if it is needed.
- A malfunction of the disk or diskpack that stores the database files causes corruption. Use rebuild or reconstruct recovery to fix the corruption. Use the command IR REBUILD or IR RECONSTRUCT or the DMSUTL utility to initiate these recoveries. Use reconstruct recovery only if certain known structures are corrupt. Otherwise use rebuild recovery.
- A logical coding error in a user program results in corrupt data in the database. Use rollback recovery or rebuild recovery to fix this corruption. The choice between rollback recovery and rebuild recovery depends on the following: Are fewer updates required to back up from the current point in time to the place where the corruption started (rollback recovery) or to move forward from a backup version of the database to the point at which the corruption occurred (rebuild recovery)? The user programs must repeat transactions that occurred after the corruption began. To do this, the user programs refer to the restart data set to determine the restart point for reprocessing.

### Program Abort Recovery

When a user program fails while it is in transaction state, the DBP must back out all transactions to the last syncpoint, as follows:

- The DBP allows all functions and transactions in progress to complete, and suspends all new requests. It unlocks all locked records (including restart data set records).
- The DBP initiates writes of the primary and secondary audit buffers to the audit trail.
- The DBP backs out all transactions from the point of failure to the last syncpoint using before images from the audit trail.
- The DBP writes all updated data buffers to disk or diskpack.
- The DBP writes a program abort recovery record to the audit trail.



After recovery is complete, the DBP restarts suspended programs. The user program receives an ABORT exception upon executing the following statements:

- **BEGIN-TRANSACTION.** When this occurs, the DBP does not place the program in transaction state.
- **END-TRANSACTION.** The last program to leave transaction state gets an ABORT exception and the DBP takes it out of transaction state.
- **CLOSE.** When this occurs, the DBP does not perform the CLOSE.

The user program that gets an ABORT exception should retrieve its restart record from the restart data set. The restart data set serves as an aid to redo any backed-out transactions.

It is important to understand that the DBP notifies only update programs of the abort recovery. Inquiry programs are not aware that an abort recovery has occurred. However, an abort recovery may affect the current record or current path pointers of an inquiry program. For example:

1. Program 1 adds a new record to a data set.
2. Program 2 reads that record.
3. Program 1 then causes an abort recovery to run which backs-out all changes program 1 made while it was in transaction state (including the addition of the new record in step 1).
4. If program 2 executes a FIND current on the data set, the current record no longer exists.

There is no explicit syntax to invoke a program abort recovery, because the DBP invokes it automatically when the system needs it.

Note that an attempt by a user program to CLOSE a database, while the user program is in transaction state, is illegal and causes a CLOSEERROR exception. The database will not be closed and the user program will remain in transaction state.

If the DBP has locked out a structure before program abort recovery begins, the structure will remain locked out and will be ignored by the recovery. You should perform a reconstruct recovery to fix the structure as soon as possible.

## **Halt/Load Recovery**

When a halt/load recovery occurs, the DBP must back out all transactions to the last syncpoint and then apply after images from two controlpoints back, as follows:

1. The DBP finds the end of the audit trail.
2. The DBP backs out all transactions from the end of the audit trail to the last syncpoint using before images from the audit trail.
3. The DBP finds the second controlpoint from the end of the audit trail. The DBP applies after images from that point to the last syncpoint in the audit trail, as necessary, to the database structure files to ensure that all transactions up to the last syncpoint are complete.





4. The DBP writes all updated data buffers to disk or diskpack.

5. The DBP writes a halt/load recovery record to the audit trail.

To begin a halt/load recovery, use the IR HALTLOAD command (see Section 5) or the DMSUTL utility (for more information, see the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*). Alternatively, if a user program OPENS a database and the IN USE flag in the control file is marked RECOVERY REQUIRED, the DBP initiates a halt/load recovery automatically. In the latter case, the DBP suspends any user program until the completion of the halt/load recovery.

Note that when the DBP finds the second controlpoint from the end of an audit trail, it considers any recovery, DBP OPEN, DBP CLOSE, or audit file change record encountered in the backwards search as two controlpoints.

If the DBP has locked out a structure before program abort recovery begins, the structure will remain locked out and will be ignored by the recovery. You should start a reconstruct recovery to fix the structure as soon as possible.

## Rebuild Recovery and Reconstruct Recovery

Rebuild recovery and reconstruct recovery automatically load a previously dumped backup version of the database from tape, unless you specify USERLD. You load, from audit file backup media, all audit files the DBP has created since the database backup. Rebuild recovery and reconstruct recovery then recover the database to the desired state by applying after images from the audit trail.

To begin rebuild recovery and reconstruct recovery, use the IR command or the DMSUTL utility.

To produce database backup tapes, use the DMSDLR utility. The *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities* describes the DMSDLR utility and the use of tapes made with other utilities.

## Rebuild Recovery

Rebuild recovery uses a full database backup that includes all structures. With rebuild recovery, the DBP moves the entire database from the backup forward in time until it reaches one of the following points, which you select:

- The end of an audit file.
- The end of the audit trail.
- The Beginning Of Job (BOJ) or End Of Job (EOJ) of a specific user program. This might be useful if you know that a particular user program has corrupted the database.
- The first syncpoint at or later than a given time.
- A particular audit file block or record.

If you select a rebuild recovery to the end of the audit trail, the DBP processes audit files until it reaches the end of the trail. At this point, the DBP displays a NO FILE message on the ODT and you must enter <mix no.> OF to terminate the recovery. To terminate this kind of rebuild before the end of the audit trail, remove an audit file that has not yet been processed and then enter <mix no.> OF when the DBP tries to open the missing file.

You can create the backup tape used for rebuild recovery with a full online or offline dump.



## Reconstruct Recovery

Reconstruct recovery brings selected structures from a backup up to date with the rest of the database. You specify the structures you want recovered. This is useful if a storage device malfunction corrupts only some of the structures.

You can create the backup tape used for reconstruct recovery with a full or partial, online or offline dump. All of the files you want to reconstruct must be on the backup tape.

There are two kinds of Reconstruct recovery: offline and online.

### Offline Reconstruct

None of the user programs can have the database OPEN while an offline reconstruct recovery is in progress. From the time you request the offline reconstruct until the time it is finished, the DBP prohibits user programs from opening the database. Any user programs that attempt to open the database will receive the error OPENERERROR - RECOVERY IN PROGRESS. If a user program has the database OPEN when you request an offline reconstruct recovery, the DBP will reject the request.

### Online Reconstruct

User programs can be updating the database while online reconstruct recovery is in progress. During this type of recovery, the DBP locks out only those structures that are being recovered. A user program that requests a database function that attempts to access or update a locked-out structure will receive the error IOERROR - STRUCTURE LOCKED OUT. Online reconstruct recovery allows recovery to occur without taking the database out of service.

### Details of Operation

Rebuild recovery automatically loads the entire database from a full database backup unless you specify USERLD, and applies after images from the audit trail up to the endpoint you specify. However, when the DBP finds a recovery audit record in the audit trail and applies before images back to the previous syncpoint, it eliminates the possibility of a partial transaction. Then, the DBP continues to apply after images forward from the recovery record. When a rebuild recovery reaches the endpoint you specify, it eliminates any partial transactions by applying before images back to the last syncpoint or the last time no user program was in transaction state. Then, the DBP opens a new audit and writes a rebuild recovery record to it.

Reconstruct recovery automatically loads from the backup the control file that belongs to the loaded structures, unless you specify USERLD. By reference to this "old" control file (XXXCNO), the DBP locates the point at which to begin the reconstruct recovery. The reconstruct recovery continues to the end of the audit trail, and then the entire database is in sync.

Online reconstruct recovery loads each structure file to be reconstructed under the assumed name XX:NNN where XXXNNN represents the name of the structure. Then, the DBP successfully locks out corresponding current structure files, removes the current files and changes the assumed names to the current names (the newly loaded files become the new current files). Because of the space the "assumed name" files use, online reconstruct recovery uses more disk space than offline reconstruct recovery.



## Rollback Recovery

Note that rollback recovery will not work to fix database corruption. It will only back out incorrectly posted transactions.

Rollback recovery is the inverse of rebuild recovery. A rollback recovery moves the entire database backward in time, by applying before images from the audit trail to the current database files. Rollback recovery is usually a faster recovery than loading the database and using a rebuild recovery. Also, rollback recovery is useful for correcting any logical errors caused by a user program that performs an update with incorrect input. A rollback recovery can be done:

- To the beginning of a specific audit file.
- To the BOJ or EOJ of a particular program, specified by program-name and date-timestamp of the program run.
- To the first syncpoint less than or equal to a specified date or time.
- To a particular Audit Block Number (ABN) and, optionally, to a particular offset within the ABN.

If the DBP has locked out a structure before program abort recovery begins, the structure will remain locked out and will be ignored by the recovery. You should perform a reconstruct recovery to fix the structure as soon as possible.

## RESTART DATA SETS

Whenever you specify the AUDIT option in a DASDL database description, you must also describe a restart data set. A restart data set provides a depository for information that enables user programs to determine the restart point for reprocessing after the DBP recovers the database.

A restart data set may be regarded as a normal standard data set with the following exceptions:

- The defining DASDL syntax is slightly different in that you must specify RESTART DATA SET.
- A restart data set must not contain embedded structures, or may not itself be embedded in another structure.
- A user program does not need to perform any explicit STORE operations on the data set.

It is recommended that a DBA, programmer, or user program adhere to the following criteria for the use of restart data sets:

- The record description should at the least contain fields for a "transaction count", indicating the number of transactions successfully carried out in a particular user program, and a program ID, uniquely associated with a particular user program.
- You should define an associated set with a key of program ID to enable a user program to retrieve specific records with that key.



- Each user program updating the database must, at the start of execution, find and lock (MODIFY) the record in the restart data set with the appropriate program ID or, if such a record does not exist, must CREATE one. From then on, as the user program successfully processes transactions, it should increment the transaction count field.
- User programs should not perform explicit STORE operations on the restart data set (except as listed below) nor should a user program retrieve records from the restart data set other than the one record containing the appropriate program ID.
- Upon successful user program completion (before database CLOSE), you should perform a BEGIN-TRANSACTION operation (with no audit) followed by a DELETE of the restart data set record and an END-TRANSACTION NO-AUDIT SYNC operation. This procedure ensures that the user program does not attempt to recover on its next execution.

The DBP performs an implicit STORE operation to the restart data set whenever a user program executes a BEGIN-TRANSACTION statement with AUDIT or an END-TRANSACTION statement with AUDIT. It is the responsibility of the user program to ensure that the transaction count field contains the appropriate value at that point. You must set AUDIT on either the BEGIN-TRANSACTION or END-TRANSACTION of every transaction, but not both. The DBP cannot restart any transaction that does not set AUDIT on BEGIN-TRANSACTION or END-TRANSACTION.





## SECTION 5

# HOW TO DUMP AND RELOAD DATABASES

### INTRODUCTION

This section describes how to dump and load backup copies of a database. It includes step by step procedures. You should periodically dump your database to ensure that you will have a clean backup version available if your database becomes corrupted. Use the DMSDLR program to create and use database backup tapes. The DMSDLR program puts your requests into a pass file that is passed to other parts of the Data Management System II (DMSII) to carry out your requests.

Initiate the DMSDLR utility as follows:

- With the IR command (described in Section 2) and one of the following options: DUMP, LOAD, REBUILD, or RECONSTRUCT.

OR

- With the DMSUTL utility (described in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*) which is a menu-based interface available at data communications terminals.

The COPY function automatically compares the files on tape and the files on disk or diskpack when you use the OFFLINE DUMP and the LOAD options. The COPY function does not do this comparison for ONLINE DUMP because the database could change between the time of the dump and the time of the comparison.

### FILES

Table 5-1 shows the database files and the actions taken on them when you request the DUMP or LOAD option or a rebuild or reconstruct recovery. In the file names, XXX represents the first three characters of the database name and XXXXXX represents the complete database name.

The COPY function will dump all of the files listed. Then, the COPY function returns to the database only those files marked Loaded in Table 5-1. If a file is not marked Loaded, you must load it separately with the SYSTEM/COPY utility.

The files marked Required must be present for the successful completion of a database DUMP. If not all of the required files are present, the COPY function will not dump any files and will stop.



**Table 5-1. Dumping and Loading Database Files**

| File                                      | Name                                  | Dumped<br>by full<br>Dump? | Required<br>for<br>Dump? | Loaded on<br>Load or<br>Rebuild? | Loaded on<br>Recon-<br>struct? |
|-------------------------------------------|---------------------------------------|----------------------------|--------------------------|----------------------------------|--------------------------------|
| Memory file                               | XXXMEM                                | Yes                        | Yes                      | Yes                              | No                             |
| Control file                              | XXXCNF                                | Yes                        | Yes                      | Yes                              | Yes                            |
| Data Definition<br>file                   | XXXDDF                                | Yes                        | Yes                      | Yes                              | No                             |
| Structure<br>files                        | XXX003-<br>XXX999                     | Yes                        | Yes                      | Yes                              | As<br>user<br>requests         |
| Statistics file                           | XXX001                                | Yes                        | No                       | Yes                              | No                             |
| Database program object<br>code (DBP)     | XXXXXX                                | Yes                        | No                       | If not present<br>on disk        | No                             |
| Database<br>tailored code file (ICM)      | XXXICM                                | Yes                        | No                       | If DBP is not<br>on disk         | No                             |
| Bind listing                              | XXXBND<br>(as<br>named<br>by<br>user) | Yes                        | No                       | No                               | No                             |
| Reorganize<br>DBP object code             | XXXREO                                | Yes                        | No                       | No                               | No                             |
| Reorganize<br>Tailored Code<br>File (ICM) | XXXRCM                                | Yes                        | No                       | No                               | No                             |



The *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation* discusses the database program (DBP) tailored code file (XXXICM). The reason you must dump it is that the DASDL compiler needs it should you ever do a REBIND of the DBP.

The bind listing is the printer backup file of the listing produced during the binding of the DBP. It is required for debugging the DBP, and Unisys requires it for support activities. At the time the binding of the DBP generates the bind listing, you should rename it XXXBND where XXX represents the first three characters of the database name. If you do not rename the bind listing when you use the DUMP option, the COPY function will not be able to locate the bind listing file.

#### NOTE

When you use the DUMP option, the COPY function does not dump or load audit files. You should archive the audit files on a regular basis with the SYSTEM/COPY utility.

## DUMPING TO TAPE

You start a dump of a database with the DMSDLR utility. Use the IR DUMP command or the DMSUTL utility to perform a dump.

Only one dump can be in progress at any one time. If you start a dump while another is in progress, the DBP will queue it until the first dump finishes.

The dump creates tapes that are in SYSTEM/COPY TAPE format.

A dump can be a full dump or a partial dump.

### Full Database Dump

A full database dump copies the entire database to a backup tape. You should do a full database dump regularly to guard against severe database corruption.

You can recover the entire database with the LOAD option and perform rebuild recovery using a full database backup. The backup could have been created by either an offline dump or an online dump. You can recover selected structures with reconstruct recovery using a full or partial database backup. The backup could have been created by either an online or an offline dump.

The names of the files dumped by a full database dump appear in Table 5-1.

### Partial Database Dump

A partial database dump copies selected structures to a backup tape. This is quicker than a full database dump. You can back up highly volatile or critical structures, and structures on less reliable media, more often than the full database. You can use a tape created by a partial dump only with reconstruct recovery.

A partial database dump puts on the tape only the structures you select and the control file (XXXCNF).

A dump can be an offline dump or an online dump.



## Offline Dump

The DUMP function performs an offline dump when the DBP is not running, or when the DBP is running but no user program has the database open for update. If a user program that updates the database is running when you start an offline dump, the DUMP function keeps the offline dump from executing until all user programs that update the database have closed the database. Then, the DUMP function inhibits all user programs that update the database from opening the database while the dump is in progress. If the DBP is not running when you start the offline dump, the DUMP function prevents the DBP from opening the database while the dump is in progress. The COPY function automatically compares the offline dump tapes with the original files during creation, to check for accuracy in file copying.

## Online Dump

An online dump copies structures to the backup tape while user programs are actively updating a database. This enables you to back up structures without taking a database out of operation. When the COPY function loads an online dump tape, you must perform recovery to repair the instabilities introduced because the database was changing while it was being dumped. You can use rebuild or reconstruct recovery depending on whether you load the entire database or specified structures. You can perform an online dump only with audited databases. If the DBP is not running when you start an online dump, the DUMP function brings up the DBP for the duration of the dump.

The DUMP function does not compare online dump tapes with the original files during creation, because the database may change between the time of the dump and the time of the comparison.

Because of special recovery requirements, you cannot use a backup tape produced by an online dump with the LOAD option. You must use the tape with rebuild or reconstruct recovery.

The recovery must encompass the entire time span during which the online dump was in progress. That is, the time you specify for the endpoint of the rebuild or reconstruct recovery must be at or after the time the online dump was completed. In this way, the DBP accounts for changes that were made to the database while the online dump was in progress.

## RECOVERY USING DATABASE BACKUPS

Use the following methods to recover a corrupted database with backup tapes: the LOAD option, rebuild recovery, or reconstruct recovery.

### LOAD Option

The LOAD option loads the entire database from a backup, including the database program (and the ICM) if it is not present on disk. It does not load the binder listing (XXXBND) or various other files that may be on the backup tape (refer to Table 5-1). The database remains stable and ready to run, in the same state it was in when it was dumped. You do not need to perform recovery and the DBP does not allow you to perform recovery.

The backup must be a full database backup produced by an offline dump.

You start the LOAD option with the IR LOAD command or the DMSUTL utility.





## **REBUILD Option**

Rebuild recovery rebuilds the entire database up to a point you specify. Rebuild recovery takes a stable version of the database from a backup tape and rebuilds it using information from the audit trail. After the recovery process rebuilds the database, it is stable and ready to run, and is left in the state it was in at the endpoint you specified. Rebuild recovery is valuable when a user program has corrupted the entire database; in this case, choose an endpoint that is before the corruption occurred. You should use rebuild recovery if you have tried reconstruct recovery but reconstruct recovery was not powerful enough to recover the database.

Use rebuild recovery to recover an entire database from a full database backup produced by an online dump. Also, you can use rebuild recovery with a full offline dump.

You start rebuild recovery with the IR REBUILD command or the DMSUTL utility.

## **RECONSTRUCT Option**

Reconstruct recovery recovers only database structures you specify, using stable versions from a database backup tape and bringing them up to date with the rest of the database. Reconstruct recovery is valuable when a storage device malfunction corrupts certain known structures. When the recovery process is finished, the database remains stable and ready to run in the state it was in before the corruption occurred.

Use reconstruct recovery to recover with a partial database backup tape. Also, you can use it with a full database backup tape. You can produce the backup tape with either an online or an offline dump.

You start the reconstruct recovery with the IR RECONSTRUCT command or the DMSUTL utility.

## **BACKING UP WITH THE SYSTEM/COPY UTILITY**

It is highly recommended that you do database backup functions through the DMSDLR utility program, with the IR command or the DMSUTL utility and the procedures in this section. DMSDLR offers the advantage of special error checking, and you can access it conveniently with the menu screens of the DMSUTL utility program. However, if you want to carry out database backup functions with the SYSTEM/COPY utility, observe the following points:

- You can load backup tapes dumped with the SYSTEM/COPY utility with the LOAD option or with rebuild or reconstruct recovery, following the rules for full and partial backup tapes that appear in this section.
- If you use the SYSTEM/COPY utility to make a full database dump, you should dump the same files as those dumped through DMSDLR. Refer to Table 5-1 for the file list.
- If you use the SYSTEM/COPY utility to make a partial database dump, you must dump the control file (XXXCNF) to the tape along with the desired structure files.



## PROCEDURES

You can use the DMSUTL utility or the IR command to dump and reload databases. The following paragraphs give step-by-step procedures for these methods.

### Using the Menu Screens (DMSUTL)

Use the DMSUTL utility to:

- Dump a full database
- Load a database that does not need recovery
- Recover a database

#### To dump a full database:

1. Mount one or more scratch tapes.
2. At a data communications terminal, start the DMSUTL utility.
3. Select DMSDLR (choice 9).
4. Select the DUMP option (choice 8).
5. When the database dump has finished, dump the audit files to a separate audit backup tape with the SYSTEM/COPY utility.

#### To load a database that does not need recovery:

1. If the most current audit file is not present on disk or pack, load the most current audit file from the audit backup tape with the SYSTEM/COPY utility. Take care not to overwrite audit files that already exist on disk or pack.
2. Mount a full database backup tape produced by an offline dump.
3. At a data communications terminal, start the DMSUTL utility.
4. Select the DMSDLR utility (choice 9).
5. Select the LOAD option (Choice 7). Enter the requested information. The database is ready to run when the backup tape finishes loading. You do not need to perform recovery and the DBP does not allow you to perform recovery.



#### **To recover a database:**

This is the only way you can use a database backup tape produced by an online dump or a partial database backup tape.

1. Load the necessary audit files from the audit backup tape with the SYSTEM/COPY utility.
2. Mount the database backup tape.
3. At a data communications terminal, start the DMSUTL utility.
4. Select the DMSDLR utility (choice 9).
5. If you are recovering the entire database, choose the REBUILD option (choice 1) and enter the requested information.

If you are recovering selected structures, choose the RECONSTRUCT option (choice 3) and enter the requested information.

#### **Using the IR Command**

Use the IR command for the following actions:

- To start a full offline dump or online dump of the entire database
- To start a partial offline or online dump of certain structures
- To load an entire database that does not need recovery
- To recover an entire database
- To recover certain structures while the database is in use
- To recover certain structures while the database is not in use

#### **To start a full offline dump of the entire database:**

1. Mount scratch tape(s).
2. Enter IR <database name> DUMP = TO <tape name>.

For duplicate database dump tapes, enter several names separated by commas in place of <tape name>.

3. Dump audit files (XXXA01 through XXXZ99, where XXX represents the first three characters of the database name) to a separate audit backup tape with the SYSTEM/COPY utility.



**To start a full online dump of the entire database:**

1. Mount scratch tape(s).
2. Enter IR <database name> DUMP = TO <tape name> ONLINE.

For duplicate tapes, enter several tape names separated by commas in place of <tape name>.

3. Dump audit files (XXXA01 through XXXZ99, where XXX represents the first three characters of the database name) to a separate audit backup tape with the SYSTEM/COPY utility.

**To start a partial offline dump of certain structures:**

1. Mount scratch tape(s).
2. Enter IR <database name> DUMP <structures to be dumped> TO <tape name>.

For duplicate database backup tapes, enter several tape names separated by commas in place of <tape name>.

3. Dump audit files (XXXA01 through XXXZ99, where XXX represents the first three characters of the database name) to a separate audit backup tape with the SYSTEM/COPY utility.

**To start a partial online dump of certain structures:**

1. Mount scratch tape(s).
2. Enter IR <database name> DUMP <structures to be dumped> TO <tape name> ONLINE.

For duplicate database backup tapes enter several tape names separated by commas in place of <tape name>.

3. Dump audit files (XXXA01 through XXXZ99, where XXX represents the first three characters of the database name) to a separate audit backup tape with the SYSTEM/COPY utility.

**To load an entire database that does not need recovery:**

1. Mount the database backup tape.
2. Enter IR <database name> LOAD = FROM <tape name>.





**To recover an entire database:**

This is a way to use a complete database backup produced by an online dump. A complete database backup can be either a full database backup tape or a database backup you created.

1. With the SYSTEM/COPY utility, load from your audit backup tape all audit files created between the time the database dump was made and the selected endpoint.

**From A Full Database Backup Tape**

2. Mount a full database backup tape.
3. To bring the database up to its most recent state, enter IR <database name> REBUILD FROM <tape name>.

You may select other endpoints for the rebuilding process; Section 2 contains the complete syntax.

**From A User Created Database Backup**

2. Restore the entire database to its media of origin. (The methods you use to do this depend on the methods you used to create the database backup.)
3. To bring the database up to its most recent state, enter IR <database name> REBUILD FROM USERLD.

You may select other endpoints for the rebuilding process; Section 2 contains the complete syntax.

**To recover certain structures while no one is using the database:**

In this procedure, you can use either a database backup tape (full or partial), or a database backup you created (full or partial).

1. Use the SYSTEM/COPY utility to load from your audit backup tape all audit files made since the database backup was made.

**From A Full Or Partial Database Backup Tape:**

2. Mount full or partial database backup tape.
3. Enter IR <database name> RECONSTRUCT <list of structures> FROM <tape name>.

**From A Full Or Partial User Created Database Backup:**

2. Restore the control file to its media of origin as XXXCNO, where XXX represents the first three characters of the database name. (The methods you use to restore portions of the database depend on the methods you used to create the database backup.)
3. Restore the desired structure files to their media of origin (using appropriate methods).
4. Enter IR <database name> RECONSTRUCT <list of structures> FROM USERLD.



**To recover certain structures while the database is in use:**

In this procedure, you can use a database backup tape (full or partial) or a database backup you created (full or partial).

1. Use the SYSTEM/COPY utility to load from your audit backup tape all audit files made since the database backup was made.

**From A Full Or Partial Database Backup Tape:**

2. Mount full or partial database backup tape.
3. Enter IR <database name> RECONSTRUCT <list of structures> FROM <tape name> ONLINE.

**From A Full Or Partial Database Backup You Created:**

2. Restore the control file to its media of origin as XXXCNO where XXX represents the first three characters of the database name. (The methods you use to restore portions of the database depend on the methods you used to create the database backup.)
3. Restore the desired structure files to their media of origin (using appropriate methods). Make sure that the restored structure files are named XX:NNN, where XX represents the first two characters of the database name and NNN represents the structure number.
4. Enter IR <database name> RECONSTRUCT <list of structures> FROM USERLD ONLINE.



# INTRODUCTION

If you do not initiate the DBP as a Production DBP, and a user program other than the bound-in one opens the database, the DMS2 Extension Module initiates the DBP exactly as if it were not a Production DBP. In this case, the bound-in user program will not execute. To initiate the bound-in user program, after the Production DBP is already running, use the same steps you did to initiate the Production DBP.

1. Compile the DASDL source in the normal manner, as described in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*. Compile the user program in the normal manner, following the instructions in the host language manual. Test the DBP and the user program, making sure they work properly together.
2. Recompile the user program with the host language compiler. Include the compiler control record \$DMSICM in the source file when compiling. This will cause the compiler to produce, instead of a complete object program, an independently compiled module (ICM) tailored for use in a Production DBP.
3. Recompile the DASDL source with the DASDL compiler. Include the compiler control record \$REBIND PRODUCTION in the source file during the compilation. This compiler control record causes the DASDL compiler to bind the user program ICM into the code it is generating to produce a Production DBP.

```

$DMSICM [ "< ICM name >" ] [ "<device>" ]

```





NOTE

You should be aware that ISAM file handling will not work within the ICM. A Production DBP cannot access ISAM files.

The ICM name becomes the name of the user program ICM. The default value is XXXPR1 where XXX represents the first three characters of the database name. The ICM name can have a maximum of six characters and must begin with an alpha character.

The device, if included, must be DISK.

Figure 6-2 shows the syntax for the REBIND PRODUCTION compiler control record.

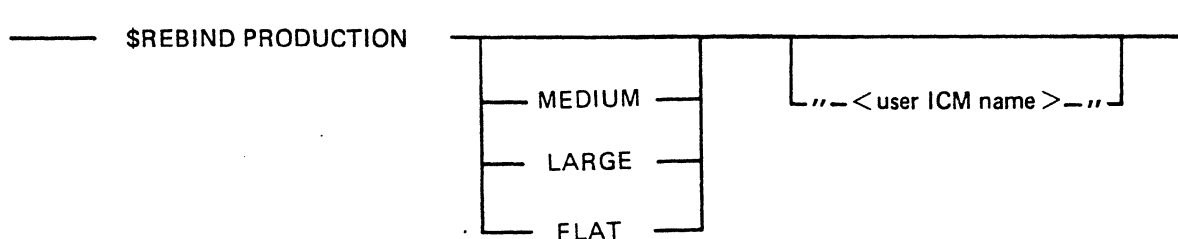


Figure 6-2. REBIND PRODUCTION Syntax

When you specify MEDIUM, LARGE, or FLAT, you choose the bind size, and hence the segmentation and overlaying characteristics, of the Production DBP. The meaning is the same as for normal DASDL binds. The default value is MEDIUM. FLAT specifies that the Production DBP contains the minimum number of overlays.

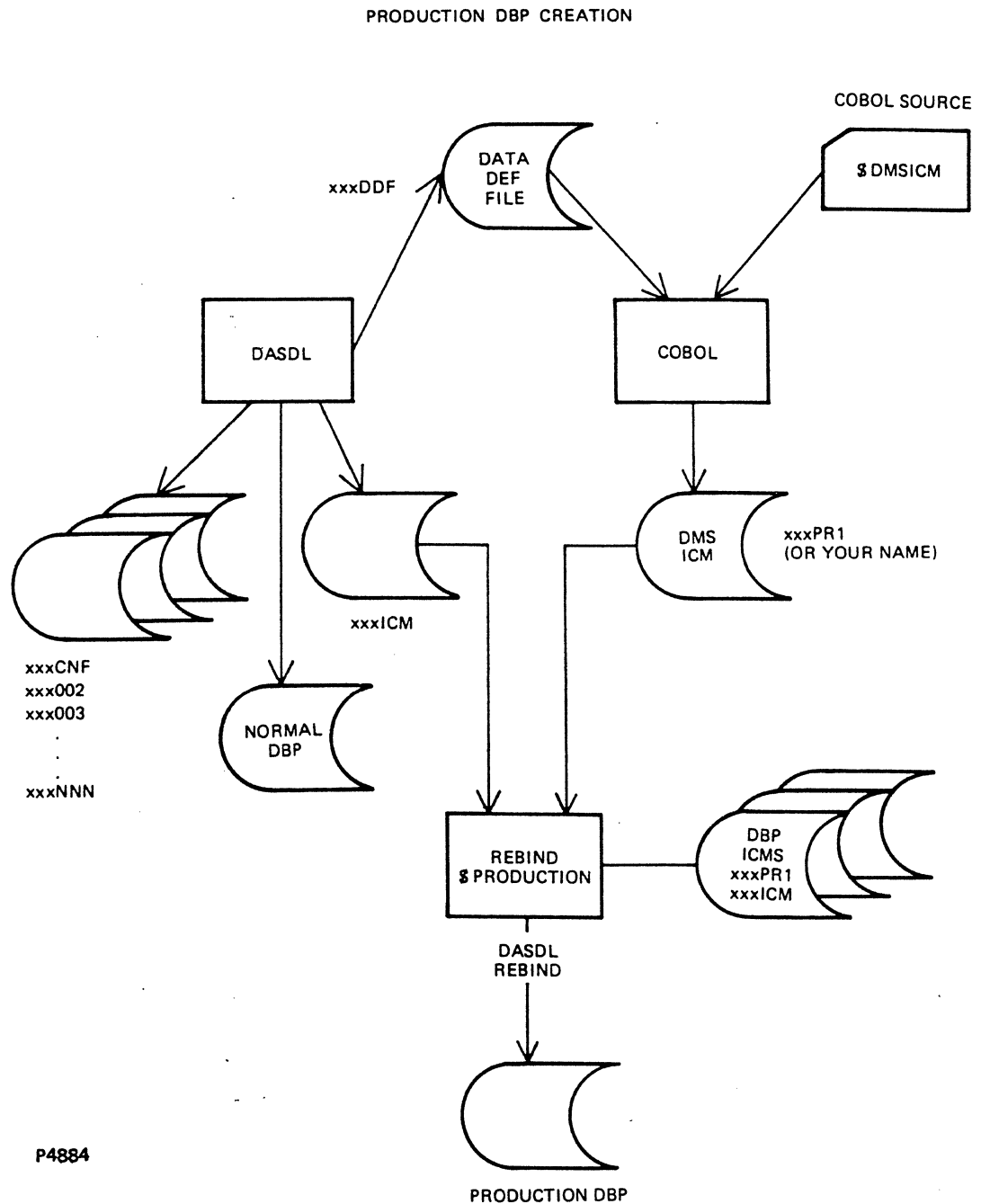
The user ICM name is the name of the ICM that the host language compiler generated (see step 2). The default value is XXXPR1, where XXX represents the first three characters of the database name.

If you include the \$CODE compiler control record in the DASDL source when you compile the Production DBP, you will get a listing of all generated code, with references to points in the compile listing produced by the host language compiler.





B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Production Database Programs



P4884

Figure 6-3. Production DBP Creation



## INITIATING THE PRODUCTION DBP

You can initiate the user program code in the Production DBP in the following ways:

- Use the IR PRODUCTION command (see Section 2).
- Use the DMSUTL utility (see the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*). Select the DMSDLR option and then the PRODUCTION option.

If the bound-in user program uses program switches, you can set the switches at initiation time by using the VALUE, SW, or IN control commands. Section 2 contains an explanation of the IR command and the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities* contains an explanation of the DMSUTL utility.

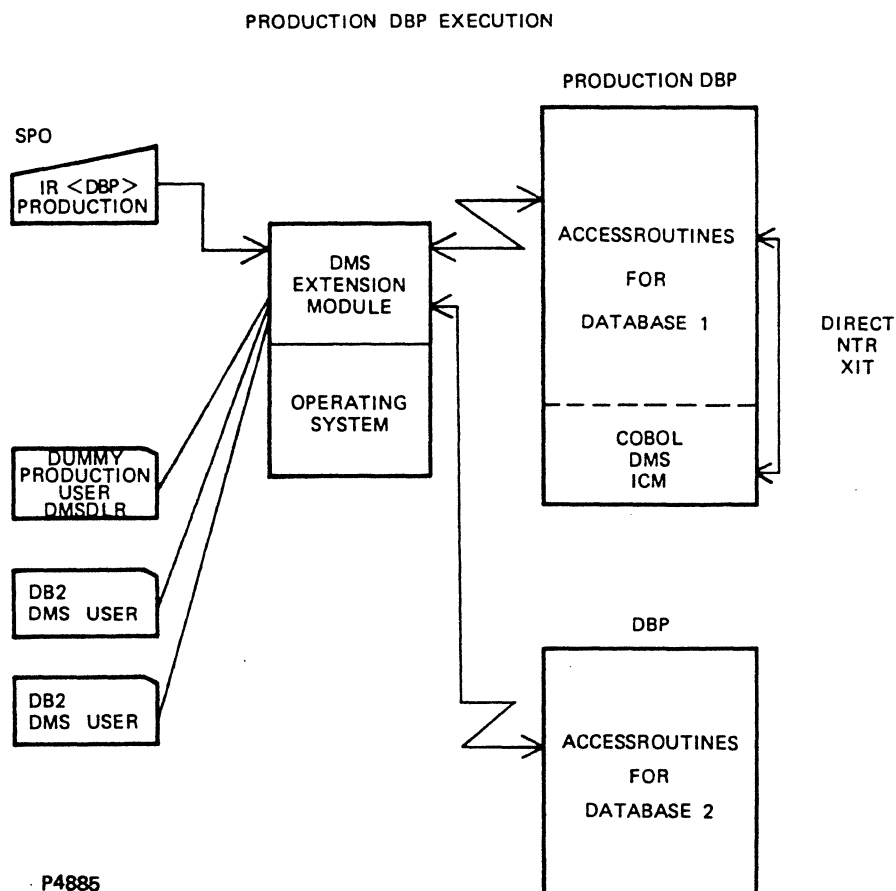


Figure 6-4. Production DBP Execution



## DETAILS

The user program ICM (DMSICM) is identical to an ICM generated using the \$ICM option except that:

- All Data Management System II (DMSII) constructs have NTR instructions to the DBP.
- All STOP RUN statements have NTR instructions to the DBP so that the database can be closed in an orderly fashion.

The only database opened in the DMSICM is the one accessed by the DBP that the DMSICM is bound to.

The DASDL compiler maintains the overlay scheme of the user program when it is bound into the Production DBP. The DASDL compiler combines the global segment of the user program with the DBP global segment; all other segments of the user program are overlaid in their own areas separately from the DBP code.

The DMSDLR utility (refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*) is the only way to initiate a Production DBP. DMSDLR opens the database and passes options to the DBP to specify production mode. The Production DBP then enters the user program routines and begins processing.

The Production DBP and DMSDLR continue processing until the user program code does a STOP RUN. This STOP RUN causes a special entry to the DBP code that specifies a terminate option. The Production DBP then closes the database and terminates normally, passing the DMSTATUS register value to DMSDLR, which then terminates.

If the DBP encounters an error in the user program code that would normally cause an abnormal termination (DS or DP) of the user program, it begins an ARM routine. The DBP ARM routine displays explanatory messages concerning the error followed by the message:

### ERROR OCCURRED IN PRODUCTION USER CODE

Use the MCP DMPOUT utility and analyze the dump. The DBPDMP analysis is of little use because it will not provide a break out of user File Information Blocks (FIBs). Whenever such an error occurs, the Production DBP closes the database before it aborts. This process ensures the integrity of the database files. Refer to the PM command in the *V Series MCP/VS System Software Operations Guide Volume 2: System Commands* for more information about the DMPOUT utility.

If the bound-in user program is waiting for something, no other user program can access the database. When the bound-in user program requests a function from the database handler section of the code, it relinquishes control. The database handler section of the code is then free to service other user programs. This situation necessitates careful planning of the user program, especially if it is a data communication program.

If a Production DBP is acting as a "normal" DBP, initiated by a normal OPEN of the database, the bound-in user program code does not execute. The Production DBP continues to act as a normal DBP until a user program opens the database with the PRODUCTION option set.

When the bound-in user program completes, DMSDLR goes to End Of Job (EOJ), and the Production DBP continues to process any outside user programs. You can re-initiate the bound-in user program after this happens, if necessary.



## SECTION 7

# SYSTEM MESSAGES AND ERROR MESSAGES FOR OPERATORS

### INTRODUCTION

The database program (DBP) and the Reorg DBP produce the messages that appear in this section. The listing is alphabetical. In many cases, the text contains suggestions for corrective action. The text indicates that certain messages apply only to Standard DBPs; this means that they do not apply to Reorg or Production DBPs. The DASDL compiler produces the error messages that appear in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*.

The DBP produces three types of messages: nonfatal and fatal error messages and information only system messages. When a nonfatal error occurs, the DBP puts out the error message and continues to execute. When a fatal error occurs, the DBP puts out the error message and terminates. Information only system messages tell you what the Data Management System II (DMSII) is doing.

### MESSAGES

ALLOWEDCORE = <integer> DIGITS

You have used the SP command to display the ALLOWEDCORE value.

ALLOWEDCORE CHANGED TO <integer> DIGITS

You have used the SP command to change the ALLOWEDCORE value.

ALLOWEDCORE SET TO <integer> DIGITS

You have used the SP command to set the ALLOWEDCORE value. This value will remain in effect for one execution of the DBP.

ALLOWEDCORE VALUE CANNOT CROSS 1MD BOUNDARY

You have used the SP command to SET the value of ALLOWEDCORE. However, the value would cause the database program to go from below one million digits (1MD) to above 1MD or vice versa. This is not allowed.

ALLOWEDCORE VALUE CANNOT EXCEED <integer>

You have used the SP command to enter a number as ALLOWEDCORE that is greater than the maximum allowable limit. When DMSII runs under MCPIX, the maximum limit is 6 million digits. Under MCP/VS 1.0 or 2.0, the maximum limit is 5,990,000 digits. Reenter the SP command with a smaller number for ALLOWEDCORE.

ALLOWEDCORE VALUE EXPECTED

You have used the SP command to change the value of ALLOWEDCORE, but you have not entered a new value. Reenter the SP command with a valid integer as the ALLOWEDCORE value.

ALLOWEDCORE VALUE LESS THAN STACKSIZE

You have used the SP command to enter a number as the ALLOWEDCORE value that is less than the current STACKSIZE value. Enter a larger number for ALLOWEDCORE or change STACKSIZE to a smaller number.





#### ALLOWEDCORE VALUE MUST BE AN INTEGER

You have used the SP command to change the ALLOWEDCORE value, but the characters you entered as the new value are not a valid number. Reenter the SP command with a valid ALLOWEDCORE value.

#### ALLOWEDCORE VALUE MUST BE GREATER THAN <integer>

You have used the SP command to enter a number as ALLOWEDCORE that is less than the minimum limit. The DASDL compiler sets the minimum limit of ALLOWEDCORE based on the \$BIND compiler control option and other factors. Reenter the SP command with a larger number for ALLOWEDCORE.

[ALREADY BEYOND SPECIFIED ABN,]  
NO RECORDS FOUND FOR REBUILD,  
HALTLOAD RECOVERY PERFORMED.

-or-

[ALREADY AHEAD OF SPECIFIED ABN,]  
NO RECORDS FOUND FOR ROLLBACK,  
HALTLOAD RECOVERY PERFORMED.

You have made an attempt to perform rebuild or rollback recovery to a stopping point that is before the current state of the database, or the rebuild or rollback recovery function did not detect any updates that were not already in the database, but did call for a halt/load recovery.

AUDIT FILE CHECKSUM ERROR  
AUDIT FILE <file name>  
AUDIT RRN <file relative block number>  
RETRY FAILED  
ALTERNATE RETRY FAILED

The DBP detected a CHECKSUM error when it was processing the designated audit file block. If you set DUPLICATE audit option, then the DBP will retry from the alternate media. Correct the media problem and retry the recovery, if necessary. Otherwise perform rebuild recovery on the database from a prior dump with a stopping point previous to the block in error. This process applies to a Standard DBP only.

AUDIT FILE CHECKSUM ERROR  
AUDIT FILE <file name>  
AUDIT RRN <file relative block number>  
RETRY SUCCESSFUL  
ALTERNATE RETRY FAILED

The DBP detected a CHECKSUM error when it was processing the designated audit file block. If you set DUPLICATE audit option, then the DBP will retry from the alternate media. Correct the media problem and retry the recovery, if necessary. Otherwise, perform rebuild recovery on the database from a prior dump with a stopping point previous to the block in error. This process applies to a Standard DBP only.



AUDIT FILE <file name> CUT OFF AT  
ABN <audit block number>, OFF <offset>

Rebuild recovery is complete to the designated stopping point. The DBP has cut off the existing audit trail at the designated point and has created a new audit file (with the next higher number) that contains only a rebuild recovery record at this point. You should remove all audit files above this point from the media because they are now invalid for the current database. However, you should have saved them with the prior dump for recovery purposes. This process applies to a Standard DBP only.

AUDIT FILE <file name> RELEASED

The DBP has closed an audit file. During normal processing, the DBP displays this message when an audit file has been filled and can be dumped to an alternate media. When this message appears at database CLOSE, file name is the last file the DBP was using at the time. If you dump the file and remove it from the media, the DBP will continue with the next sequential file on the next database OPEN. Otherwise, the DBP will continue to use the file until it is filled. During recovery, the message indicates that the DBP has finished processing the records on that file; however, it can return to the file depending on the type of recovery in progress.

AUDIT FILE NOT OPEN — NO UPDATING USERS

You have used the SP command to: 1) close a database's current audit trail file(s) and open the next sequential audit trail file(s) or 2) display the name of the current audit file. The audit file, however, is not open because no user programs are currently updating the database. Check to be sure that you have entered the database name correctly.

AUDIT FILE NOT SWITCHED — CURRENT FILE EMPTY

You have used the SP command to invoke the AUDIT FILE SWITCH routine. The current audit file has not been written to so there is no reason to switch files.

AUDIT IS [NOT] SET

The Reorg DBP displays this information only message.

AUDIT OPTION SET

PRIMARY AUDIT FILE: <audit file> ABN: <integer>

You have used the SP command to display the status of the AUDIT option.

BEG COMPRESS <data set structure name>

This information only message comes from the Reorg DBP. It indicates that the compression process is beginning.

BEG COMPRESS <data set structure name> ORDERED BY <set structure name>

This information only message comes from the Reorg DBP. It indicates that the compression process is beginning and that the data set will be ordered by the specified set.

BEG COPY BACK <structure name>

This information only message comes from the Reorg DBP. It indicates that the COPY BACK function is beginning.

BEG FIXUP <master data set structure name>

This information only message comes from the Reorg DBP.



BEG FIXUP <set/subset structure name>

This information only message comes from the Reorg DBP.

BEG GENERATE <set/subset structure name>

This information only message comes from the Reorg DBP. It indicates that the generate process is beginning.

BEGINNING <recovery type>

This information only message indicates that the denoted recovery process is beginning.

Recovery type can be:

ABORT RECOVERY  
HALTLOAD RECOVERY  
OFFLINE RECONSTRUCT RECOVERY  
ONLINE RECONSTRUCT RECOVERY  
PURGE RECOVERY  
REBUILD RECOVERY  
ROLLBACK RECOVERY

BUFFER TOTAL EXCEEDS 99, BUFFERS NOT MODIFIED

You have used the SP command to change or to set the buffers allocated to a given structure. The new allocation, however, would bring the total number of buffers to more than 99. If a change results in more than 99 buffers, the change does not take effect. Reenter the SP command with different buffer allocations that will produce less than 99 total buffers.

BUFFER TYPE EXPECTED: (<integer> SYSTEM/USER/RANDOM/SERIAL)

You have entered an incorrect buffer type with the SP command. Reenter the SP command with one of the four valid buffer types.

BUFFER VALUE MUST BE AN INTEGER

You have used the SP command to change the number of buffers allocated to a given structure, but you have entered non-numeric characters in a place where you should enter the number of buffers. Reenter the SP command with numeric characters.

BUFFER VALUE OUT OF RANGE (0 - 99)

At least one of the buffer allocations you entered with the SP command (SYSTEM, USER, RANDOM and so on) is greater than 99. Reenter the SP command with valid buffer allocations.

BUFFERS FOR <structure name>

SYSTEM: <nn> USER: <nn> SERIAL: <nn>

You have used the SP command to display the current buffer information about structure name.

BUFFERS FOR <structure name> CHANGED TO

SYSTEM: <nn> USER: <nn> SERIAL: <nn>

You have used the SP command to change the buffer allocation of structure name to the displayed values.



**BUFFERS FOR <structure name> SET TO  
SYSTEM: <nn> USER: <nn> SERIAL: <nn>**

You have used the SP command to set the buffer allocations of structure name to the displayed values. These values remain in effect for one execution of the DBP.

**CAN'T REFERENCE ACCESS STRUCTURE**

You have used the SP command to display, set or change the parameters of a given structure, but the structure name you entered is that of an access. Accesses cannot be displayed or changed with the SP command because they are logical, not physical, structures. Reenter the SP command with the name of an existing data set, set, or subset.

**CAN'T REFERENCE DELETED STRUCTURE**

You have used the SP command to display, set or change the parameters of a structure, but the structure you entered has previously been deleted from the database. Check to be sure that you entered the structure name correctly. Reenter the SP command with the name of a valid data set, set or subset.

**CAN'T REFERENCE REMAP STRUCTURE**

You have used the SP command to display, set or change the parameters of a structure, but the structure name you entered is that of a remap structure. Remap structures are logical, not physical; there are no buffers or writeaheads for them. The SP command cannot operate on logical structures. Reenter the SP command with the name of a valid data set, set or subset.

**CHANGE OPTION INVALID**

You have used the SP command to change at least one database parameter, but the characters you entered are not one of the keywords for the parameters that can be changed (SYNCPOINT, ALLOWEDCORE, and so on). Usually, this message indicates that you have entered a keyword incorrectly. Reenter the SP command with correct keywords and values.

**CNF LOCK SHOULD BE: <lock type>**

The DBP has discovered an invalid lock state in the control file during an attempt to run database recovery. This message applies to the Standard DBP only. If this message appears, load a previous dump of the database and perform rebuild recovery.

The lock type can be one of the following:

NOT LOCKED OR DBP LOCK  
DBP LOCK OR HALT LOAD RUNNING  
NO LOCK, DBP LOCK, OR REBUILD RUNNING  
NO LOCK, DBP LOCK, OR ROLLBACK RUNNING  
NOT LOCKED, DBP LOCK, OR RECONSTRUCT RUNNING

**CONTROLPOINT = <integer> SYNCPOINTS**

You have used the SP command to display the number of syncpoints per controlpoint.

**CONTROLPOINT CHANGED TO <integer> SYNCPOINTS**

You have used the SP command to change the number of syncpoints per controlpoint.

**CONTROLPOINT SET TO <integer> SYNCPOINTS**

You have used the SP command to set the number of syncpoints per controlpoint to the displayed value. This value will remain in effect for one execution of the DBP.





#### CONTROLPOINT VALUE EXPECTED

You have used the SP command to change the number of syncpoints per controlpoint, but you have not entered a new CONTROLPOINT value. Reenter the SP command with a valid number for the CONTROLPOINT value.

#### CONTROLPOINT VALUE MUST BE AN INTEGER

You have used the SP command to change the CONTROLPOINT value, but the characters you entered as the new value are not a valid number. Reenter the SP command with a valid CONTROLPOINT value.

#### CONTROLPOINT VALUE OUT OF RANGE (1 - 9999)

You have used the SP command to enter a CONTROLPOINT value that is outside of the allowed range. The number of syncpoints per controlpoint cannot be less than 1 or greater than 9999. Reenter the SP command with a valid value for CONTROLPOINT.

#### CONTROL FILE CORRUPTED BY PURGE FAILURE. REBUILD REQ'D

The DBP displays this message at the time of database OPEN. The DBP crashed previously during a PURGE function, corrupting the control file. It is necessary to do a rebuild recovery from a known stable state previous to the failed PURGE function. Refer to Section 3 for more information about the PURGE function.

#### DATABASE AUTOLOAD LOAD FROM <tape name> INITIATED

You have initiated DMSDLR with the LOAD FROM <tape name> function requested.

#### DATABASE AUTOLOAD REBUILD ABORTED:

PASS FILE FROM DMSDLR (<file name>) NOT FOUND.

You have initiated DMSDLR with the REBUILD FROM <tape name> function requested. DMSDLR calls the COPY function to carry out the actual file transfers. DMSDLR builds a file that contains syntax for the COPY function. This file is missing from the disk. No files have been transferred and the rebuild recovery has not been done. Use the RESTART function to recover.

#### DATABASE AUTOLOAD REBUILD FROM <tape name> ABORTED:

ANOTHER MANUAL LOAD OR REBUILD/RECONST IS ALREADY RUNNING.

This error message concerns the rebuild recovery process that you initiated with the IR REBUILD command or through the REBUILD option of the DMSUTL utility. Wait until the current recovery is finished, and then reenter the REBUILD request.

#### DATABASE AUTOLOAD REBUILD FROM <tape name> ABORTED:

DATABASE EXCEPTION RETURNED ON REBUILD.

ERROR CAT: <category number> SUBCAT: <subcategory number>

You have requested that a database backup tape be reloaded to the database, by initiating the DMSDLR utility with the REBUILD FROM <tape name> function requested. DMSDLR initiates the DBP to carry out the rebuild recovery. An error has occurred in a DBP operation relating to the recovery. An explanation of the error category and subcategory appears in Section 8.

#### DATABASE AUTOLOAD REBUILD FROM <tape name> ABORTED:

FILES CANNOT BE LOADED WHILE THE DBP IS RUNNING.

You have initiated DMSDLR with the REBUILD FROM <tape name> function requested. Database files cannot be loaded while the database is open. This message indicates that the request was rejected because the DBP had the database open. No files have been loaded from the backup tape and the rebuild recovery has not been done. Try again when the DBP is not running.



DATABASE AUTOLOAD REBUILD FROM <tape name> ABORTED:  
FUNCTION DID NOT COMPLETE SUCCESSFULLY.

You have initiated DMSDLR with the REBUILD FROM <tape name> function requested. DMSDLR calls the SYSTEM/COPY utility to do the actual file transfer, and calls the DBP to do the rebuild recovery. An error has occurred in loading the tape. Some files may have been loaded. Use the RESTART function to recover.

DATABASE AUTOLOAD REBUILD FROM <tape name> ABORTED:  
THE TAPE IS NOT AN ACCEPTABLE DATABASE DUMP TAPE.

This error message concerns the REBUILD function of the DMSDLR utility that you initiated with the IR REBUILD command or through the REBUILD option of the DMSUTL utility. The REBUILD function accepts only database backup tapes created with DMSDLR or the SYSTEM/COPY utility. No files have been transferred. Mount a proper database backup tape and repeat the REBUILD request.

DATABASE AUTOLOAD REBUILD FROM <tape name> ABORTED:  
THIS FUNCTION IS NOT ALLOWED WHILE THE DBP IS RUNNING.

This error message concerns the REBUILD function of DMSDLR utility that you initiated with the IR REBUILD command or through the REBUILD option of the DMSUTL utility. Only ONLINE RECONSTRUCT recovery can be done while the database is open. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE AUTOLOAD REBUILD FROM <tape name> ABORTED:  
THIS FUNCTION NOT ALLOWED WITH PARTIAL DATABASE DUMPS.

This error message concerns the REBUILD function of DMSDLR utility that you initiated with the IR REBUILD command or through the REBUILD option of the DMSUTL utility. You can use the REBUILD function only with full database backup tapes. Use the RECONSTRUCT function to load a partial database backup tape. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE AUTOLOAD REBUILD FROM <tape name> COMPLETE.

This is an information only message about the rebuild function of DMSDLR utility.

DATABASE AUTOLOAD REBUILD FROM <tape name> INITIATED.

You have initiated DMSDLR with the REBUILD FROM <tape name> function requested.

DATABASE AUTOLOAD RECONST FROM <tape name> ABORTED:  
ANOTHER MANUAL LOAD OR REBUILD/RECONST IS ALREADY RUNNING.

This error message concerns the RECONSTRUCT function of DMSDLR utility that you initiated with the IR RECONSTRUCT or RECONSTRUCT option of DMSDLR utility. Wait until the current recovery procedure has finished; then reenter the RECONSTRUCT request.

DATABASE AUTOLOAD RECONST FROM <tape name> ABORTED:  
FILE DUMPING/LOADING DID NOT COMPLETE SUCCESSFULLY.

Use IR RESTART command or the RESTART option of the DMSUTL utility to recover.



DATABASE AUTOLOAD RECONST FROM <tape name> ABORTED:  
ALL SPECIFIED FILES MUST BE ON THE DATABASE DUMP TAPE.

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. The files that you requested to be loaded are not all present on the backup tape. The DBP has ignored the request. No files have been loaded from the backup tape and the reconstruct recovery has not been done. Reenter the reconstruct recovery request with a different file list, or use a different backup tape and reenter the reconstruct recovery request.

DATABASE AUTOLOAD RECONST FROM <tape name> ABORTED:  
FILES CANNOT BE LOADED WHILE THE DBP IS RUNNING.

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. Database files cannot be loaded while the database is open. This message indicates that the request was rejected because the DBP had the database open. No files have been loaded from the backup tape, and the reconstruct recovery has not been done. Try again when the DBP is not running or try an ONLINE RECONSTRUCT.

DATABASE AUTOLOAD RECONST ABORTED:  
PASS FILE FROM DMSDLR (<file name>) NOT FOUND.

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. DMSDLR calls the COPY function to carry out the actual file transfers. DMSDLR builds a file that contains syntax for the COPY function. This file is missing from disk. No files have been transferred, and the reconstruct recovery has not been done. Use the RESTART option to recover.

DATABASE AUTOLOAD RECONST FROM <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON RECONSTRUCT.  
ERROR CAT: <category exception> SUBCAT: <subcategory number>

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. DMSDLR initiates the DBP to carry out the reconstruct recovery. An error has occurred in DBP operation. An explanation of the error category and subcategory appears in Section 8.

DATABASE AUTOLOAD RECONST FROM <tape name> ABORTED:  
FUNCTION DID NOT COMPLETE SUCCESSFULLY.

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. DMSDLR calls the COPY function to do the actual file transfer, and calls the DBP to do the reconstruct recovery. An error has occurred in loading the tape. Some files may have been loaded. Use the RESTART option to recover.

DATABASE AUTOLOAD RECONST FROM <tape name> ABORTED:  
THE TAPE IS NOT AN ACCEPTABLE DATABASE DUMP TAPE.

This error message concerns the reconstruct recovery that you initiated with the IR RECONSTRUCT command or through the RECONSTRUCT option of the DMSUTL utility. Reconstruct recovery will accept only database backup tapes created with the DUMP function or the SYSTEM/COPY utility. No files have been loaded. Mount a valid tape and repeat the RECONSTRUCT request.

DATABASE AUTOLOAD RECONST FROM <tape name> ABORTED:  
THIS FUNCTION IS NOT ALLOWED WHILE THE DBP IS RUNNING.

The only recovery the DMS2 Extension Module allows while the DBP is running is online reconstruct recovery. Refer to Section 5 for more information about how to dump and reload a database.



**DATABASE AUTOLOAD RECONST FROM <tape name> COMPLETED.**

This is an information only message about reconstruct recovery. The DBP has loaded all files and has completed reconstruct recovery.

**DATABASE AUTOLOAD RECONST FROM <tape name> INITIATED.**

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. This is an information only message.

**DATABASE IS IN USE; TRY REORGANIZE LATER**

This is an information only message about the reorganization process.

**DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
ANOTHER MANUAL LOAD OR REBUILD RECONST IS ALREADY RUNNING.**

This error message concerns the LOAD function of DMSDLR utility that you initiated with IR LOAD command or through LOAD option of the DMSUTL utility. Wait until current recovery finishes; then reenter the LOAD request.

**DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
FILE DUMPING/LOADING DID NOT COMPLETE SUCCESSFULLY.**

Use the IR RESTART command or the RESTART option of the DMSUTL utility to recover.

**DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
FILES CANNOT BE LOADED WHILE THE DBP IS RUNNING.**

You have initiated DMSDLR with the LOAD FROM <tape name> function requested. Database files cannot be loaded while the database is open. This message indicates that the request was rejected because the DBP had the database open. No files have been loaded from the backup tape. Try again when the DBP is not running.

**DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
FUNCTION DID NOT COMPLETE SUCCESSFULLY.**

You have initiated DMSDLR with the LOAD FROM <tape name> function requested. DMSDLR calls the COPY function to do the actual file transfer. An error has occurred in loading the tape. Some files may have been loaded. Use the RESTART function to recover.

**DATABASE MANUAL LOAD ABORTED:  
PASS FILE FROM DMSDLR (<file name>) NOT FOUND.**

You have initiated DMSDLR with the LOAD FROM <tape name> function requested. DMSDLR calls the COPY function to carry out the actual file transfers. DMSDLR builds a file that contains syntax for the COPY function. This file is missing from disk. No database files have been transferred. Use the RESTART function to recover.

**DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
ANOTHER MANUAL LOAD OR REBUILD RECONST IS ALREADY RUNNING.**

Wait for the LOAD or REBUILD function to finish; then reenter the LOAD request.





DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
THE TAPE IS NOT AN ACCEPTABLE DATABASE DUMP TAPE.

This error message concerns the LOAD function of the DMSDLR utility that you initiated with the IR command or through the DMSUTL utility. You can only use LOAD with a full database dump tape produced by an offline dump or with the SYSTEM/COPY utility. Mount such a tape and repeat the LOAD request. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
THIS FUNCTION NOT ALLOWED WHILE THE DBP IS RUNNING.

This error message concerns the LOAD function of the DMSDLR utility that you initiated with the IR LOAD command or through the LOAD option of the DMSUTL utility. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
THIS FUNCTION NOT ALLOWED WITH ONLINE DATABASE DUMPS.

This error message concerns the LOAD function of the DMSDLR utility that you initiated with the IR LOAD command or through the LOAD option of the DMSUTL utility. A LOAD is allowed only from a full database dump tape produced by an offline dump. To load a full database dump tape produced by an online dump, use the REBUILD function. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE MANUAL LOAD FROM <tape name> ABORTED:  
THIS FUNCTION NOT ALLOWED WITH PARTIAL DATABASE DUMPS.

This error message concerns the LOAD function of the DMSDLR utility that you initiated with the IR LOAD command or through the LOAD option of the DMSUTL utility. A LOAD is allowed only from a full database backup tape. To load structures from a partial database backup tape, use the RECONSTRUCT function. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE MANUAL LOAD FROM <tape name> COMPLETE.

This is an information only message about the Load function of DMSDLR utility. All files have been transferred.

DATABASE MANUAL LOAD FROM <tape name> INITIATED.

You have initiated DMSDLR with the LOAD FROM <tape name> function requested. This is an information only message.

DATABASE NOT AUDITED

You have used the SP command to manipulate one of several database parameters, but the command has not taken effect because the database is not audited. Several parameters of the SP command can only operate on audited databases (SYNCPPOINT, CONTROLPOINT and so on). Check to be sure that you entered the database name correctly. Reenter the SP command with the name of an audited database.

DATABASE OFFLINE DUMP ABORTED:  
PASS FILE FROM DMSDLR (<file name>) NOT FOUND.

You have initiated DMSDLR with the DUMP TO <tape name> function requested. DMSDLR calls the COPY function to carry out the actual file transfers. DMSDLR builds a file that contains syntax for the COPY function. This file is missing from disk. No files have been transferred. Repeat the DUMP instruction.



DATABASE OFFLINE DUMP TO <tape name> ABORTED:  
ALL REQUIRED DATABASE FILES MUST BE PRESENT.

You have requested that the database be backed up to tape by initiating the DMSDLR utility with the DUMP TO <tape name> function requested. DMSDLR calls the COPY function to do the actual file transfers. To dump a database, the control file (XXXCNF), Data Definition File (XXXDDF), memory file (XXXMEM), and all structure files (XXX003-XXX999 if they exist) must be present on the disk or diskpack as specified in their physical attributes description in DASDL. The COPY function has been terminated. When the situation has been corrected, repeat the DUMP instruction. See Section 5 for more information about how to dump and reload a database.

DATABASE OFFLINE DUMP TO <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON DATABASE CLOSE.  
ERROR CAT: <category number> SUBCAT: <subcategory number>

You have requested that a database backup tape be made, by initiating the DMSDLR utility with the DUMP TO <tape name> function requested. As part of the dumping procedure, the DUMP function informs the DBP that the files are being dumped, so that no programs that update the database are allowed to open the database during the dump. An error has occurred in DBP operation relating to the dump. An explanation of the error category and subcategory appears in Section 8. If you request an online dump, a user program can update the database while the dump is going on. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE OFFLINE DUMP TO <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON DATABASE OPEN.  
ERROR CAT: <category number> SUBCAT: <subcategory number>

You have requested that a database backup tape be made, by initiating the DMSDLR utility with the DUMP TO <tape name> function requested. As part of the dumping procedure, the DUMP function informs the DBP that the files are being dumped, so that no programs that update the database are allowed to open the database during the dump. An error has occurred in DBP operation relating to the dump. An explanation of the error category and subcategory appears in Section 8. If you request an online dump, a user program can update the database while the dump is going on. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE OFFLINE DUMP TO <tape name> ABORTED:  
FILE DUMPING/LOADING DID NOT COMPLETE SUCCESSFULLY.

Use IR RESTART command or the RESTART option of the DMSUTL utility to recover.

DATABASE OFFLINE DUMP TO <tape name> ABORTED:  
STRUC FILE <file name> IS LOCKED OUT.

You have requested that a database backup be made by initiating DMSDLR with the DUMP TO <tape name> option. A database structure file is locked out. Recover the structure with a reconstruct recovery and then try making the backup again.

DATABASE OFFLINE DUMP TO <tape name> ABORTED:  
STRUC FILE <file name> NOT RECOVERED FROM PREVIOUS ONLINE DUMP.

You have requested that a database backup be made by initiating DMSDLR with the DUMP TO <tape name> option. If you load a structure from a backup tape that was created with an online dump, you must perform a recovery procedure on the structure before it can be dumped again. This has not been done. After you do the required recovery procedure, repeat the DUMP instruction.



DATABASE OFFLINE DUMP TO <tape name> COMPLETED.

This is an information only message about the Dump function. All files have been dumped.

DATABASE OFFLINE DUMP TO <tape name> INITIATED.

This is an information only message about the Dump function.

DATABASE OFFLINE PARTIAL DUMP TO <tape name> ABORTED:  
FILE DUMPING/LOADING DID NOT COMPLETE SUCCESSFULLY.

Use IR RESTART command or the RESTART option of the DMSUTL utility to recover.

DATABASE OFFLINE PARTIAL DUMP ABORTED:  
PASS FILE FROM DMSDLR (<file name>) NOT FOUND.

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. DMSDLR calls the COPY function to carry out the actual file transfers. DMSDLR builds a file that contains syntax for the COPY function. This file is missing from disk. No files have been transferred. Repeat the DUMP instruction.

DATABASE OFFLINE PARTIAL DUMP TO <tape name> ABORTED:  
ALL REQUIRED DATABASE FILES MUST BE PRESENT.

You have requested that the database be backed up to tape by initiating the DMSDLR utility with the DUMP function requested and a list of structures specified. The COPY function was called to transfer the structure files. Not all of the structure files specified are present. The COPY function has been discontinued. The tape as it exists is not usable. Correct the structure list or the missing file situation and repeat the DUMP request. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE OFFLINE PARTIAL DUMP TO <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON DATABASE CLOSE.  
ERROR CAT: <category number> SUBCAT: <subcategory number>

You have requested that a database backup tape be made by initiating the DMSDLR utility with the DUMP TO <tape name> option. As part of the dumping procedure, the DUMP function informs the DBP that the files are being dumped, so that no programs that update the database are allowed to open the database during the dump. An error has occurred in DBP operation relating to the dump. An explanation of the error category and subcategory appears in Section 8. If you request an online dump, user programs can still open the database while it is being dumped. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE OFFLINE PARTIAL DUMP TO <tape name> ABORTED:  
STRUC FILE <file name> IS LOCKED OUT.

You have initiated DMSDLR with the DUMP option. A database structure file is locked out. Recover the structure with a reconstruct recovery and then try the dump again.

DATABASE OFFLINE PARTIAL DUMP TO <tape name> ABORTED:  
STRUC FILE <file name> NOT RECOVERED FROM PREVIOUS ONLINE DUMP.

You have initiated DMSDLR with the DUMP function requested. If a structure is loaded from a backup tape that was created by an online dump, the structure must have a recovery procedure done on it before it can be dumped again. This has not been done. After you do the required recovery procedure, repeat the DUMP instruction.



DATABASE OFFLINE PARTIAL DUMP TO <tape name> COMPLETED.

This is an information only message about the Dump function of DMSDLR utility. All specified files have been dumped.

DATABASE OFFLINE PARTIAL DUMP TO <tape name> INITIATED.

This is an information only message about the Dump function of DMSDLR utility.

DATABASE ONLINE AUTOLOAD LOAD FROM <tape name> COMPLETED.

This is an information only message that indicates the LOAD phase of the online rebuild or reconstruct recovery is complete.

DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> ABORTED:  
ANOTHER MANUAL LOAD OR REBUILD/RECONST IS ALREADY RUNNING.

Wait for the current recovery to finish, then reenter the RECONSTRUCT request.

DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> ABORTED:  
ALL SPECIFIED FILES MUST BE ON THE DATABASE DUMP TAPE.

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. The files that were requested to be loaded are not all present on the backup tape. The request has been ignored. No files have been loaded from the backup tape, and the reconstruct recovery has not been done. Reenter the reconstruct request with a different file list, or use a different backup tape and reenter the reconstruct request.

DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> ABORTED:  
FILES CANNOT BE LOADED WHILE THE DBP IS RUNNING.

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. Database files cannot be loaded while the database is open. This message indicates that the request was rejected because the DBP had the database open. No files have been loaded from the backup tape, and the reconstruct recovery has not been done. Try again when the DBP is not running or try an online reconstruct recovery.

DATABASE ONLINE AUTOLOAD RECONST ABORTED:  
PASS FILE FROM DMSDLR (<file name>) NOT FOUND.

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. DMSDLR calls the COPY function to carry out the actual file transfers. DMSDLR builds a file that contains syntax for the COPY function. This file is missing from disk. No files have been transferred, and the reconstruct recovery has not been done. Use the RESTART option to recover.

DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON RECONSTRUCT.  
ERROR CAT: <category exception> SUBCAT: <subcategory number>

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> function requested. DMSDLR initiates the DBP to carry out the reconstruct recovery. An error has occurred in DBP operation. An explanation of the error category and subcategory appears in Section 8.

DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> ABORTED:  
FILE DUMPING/LOADING DID NOT COMPLETE SUCCESSFULLY.

Use IR RESTART command or the RESTART option of the DMSUTL utility to recover.





DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> ABORTED:  
FUNCTION DID NOT COMPLETE SUCCESSFULLY.

You have initiated DMSDLR with the RECONSTRUCT FROM <tape name> recovery requested. DMSDLR calls the COPY function to do the actual file transfer, and calls the DBP to do the reconstruct recovery. An error has occurred in loading the tape. Some files may have been loaded. Use the RESTART option to recover.

DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> ABORTED:  
THE TAPE IS NOT AN ACCEPTABLE DATABASE DUMP TAPE

This error message concerns the RECONSTRUCT recovery process of the DMSDLR utility that you initiated with an IR RECONSTRUCT command or by repeating the RECONSTRUCT request.

DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> COMPLETED.

This is an information only message.

DATABASE ONLINE AUTOLOAD RECONST FROM <tape name> INITIATED.

This is an information only message.

DATABASE ONLINE DUMP ABORTED:  
PASS FILE FROM DMSDLR (<file name>) NOT FOUND.

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. DMSDLR calls the COPY function to carry out the actual file transfers. DMSDLR builds a file that contains syntax for the COPY function. This file is missing from disk. No files have been transferred. Repeat the DUMP instruction.

DATABASE ONLINE DUMP TO <tape name> ABORTED:  
ALL REQUIRED DATABASE FILES MUST BE PRESENT.

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. DMSDLR calls the COPY function to do the actual file transfers. To dump a database, the control file (XXXCNF), Data Definition File (XXXDDF), memory file (XXXMEM), and all structure files (XXX003-XXX512 if they exist) must be present on the disk or diskpack as specified in their physical attributes description in DASDL. One or more of these files is missing. The COPY function has been discontinued. When the situation has been corrected, repeat the DUMP instruction. See Section 5 for more information about how to dump and reload a database.

DATABASE ONLINE DUMP TO <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON DATABASE CLOSE.  
ERROR CAT: <category number> SUBCAT: <subcategory number>

You have requested that a database backup tape be made by initiating the DMSDLR utility with the DUMP TO <tape name> function requested. As part of the dumping procedure, the DUMP function informs the DBP that the files are being dumped, so that audit records can be inserted into the audit trail. This process enables the DBP to perform proper database recovery when the online dump tape is reloaded into the database. An error has occurred in DBP operation relating to the dump. An explanation of the error category and subcategory appears in Section 8.



DATABASE ONLINE DUMP TO <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON DATABASE OPEN.  
ERROR CAT: <category number> SUBCAT: <subcategory number>

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. As part of the dumping procedure, the DUMP function informs the DBP that the files are being dumped, so that audit records can be inserted into the audit trail. This process enables the DBP to perform proper database recovery when the online dump tape is reloaded into the database. An error has occurred in DBP operation relating to the dump. An explanation of the error category and subcategory appears in Section 8.

DATABASE ONLINE DUMP TO <tape name> ABORTED:  
FILE DUMPING/LOADING DID NOT COMPLETE SUCCESSFULLY.

Use the IR RESTART command or the RESTART option of DMSUTL utility to recover.

DATABASE ONLINE DUMP TO <tape name> ABORTED:  
STRUC FILE <file name> IS LOCKED OUT

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. A database structure file is locked out. Recover the file using reconstruct recovery and then try the dump again.

DATABASE ONLINE DUMP TO <tape name> ABORTED:  
STRUC FILE <file name> NOT RECOVERED FROM PREVIOUS ONLINE DUMP.

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. If a structure is loaded from a backup tape that was created by an online dump, the structure must have a recovery procedure done on it before it can be dumped again. After you do the required recovery procedure, repeat the DUMP instruction.

DATABASE ONLINE DUMP TO <tape name> COMPLETED.

This is an information only message about the Dump function of DMSDLR utility. All files have been dumped.

DATABASE ONLINE DUMP TO <tape name> INITIATED.

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. This is an information only message.

DATABASE ONLINE PARTIAL DUMP TO <tape name> ABORTED:  
ALL REQUIRED DATABASE FILES MUST BE PRESENT.

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. The COPY function was called to transfer the structure files. Not all of the structure files specified are present. The COPY function has been discontinued. The tape as it exists is not usable. Correct the structure list or the missing file situation and repeat the DUMP request. Refer to Section 5 for more information about how to dump and reload a database.

DATABASE ONLINE PARTIAL DUMP ABORTED:  
PASS FILE FROM DMSDLR (<file name>) NOT FOUND.

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. DMSDLR calls the COPY function to carry out the actual file transfers. DMSDLR builds a file that contains syntax for the COPY function. This file is missing from disk. No files have been transferred. Repeat the DUMP instruction.



DATABASE ONLINE PARTIAL DUMP TO <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON DATABASE CLOSE.  
ERROR CAT: <category number> SUBCAT: <subcategory number>

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. As part of the dumping procedure, the DUMP function informs the DBP that the files are being dumped, so that audit records can be inserted into the audit trail. This process enables the DBP to perform proper database recovery when the online dump tape is reloaded into the database. An error has occurred in DBP operation relating to the dump. An explanation of the error category and subcategory appears in Section 8.

DATABASE PARTIAL ONLINE DUMP TO <tape name> ABORTED:  
DATABASE EXCEPTION RETURNED ON DATABASE OPEN.  
ERROR CAT: <category number> SUBCAT: <subcategory number>

You have requested that a database backup be made by initiating DMSDLR with the DUMP option. As part of the dumping procedure, the DUMP function informs the DBP that the files are being dumped, so that audit records can be inserted into the audit trail. This process enables the DBP to perform proper database recovery when the online dump tape is reloaded into the database. An error has occurred in DBP operation relating to the dump. An explanation of the error category and subcategory appears in Section 8.

DATABASE ONLINE PARTIAL DUMP TO <tape name> ABORTED:  
FILE DUMPING/LOADING DID NOT COMPLETE SUCCESSFULLY.

Use the IR RESTART command or the RESTART option of the DMSUTL utility to recover.

DATABASE ONLINE PARTIAL DUMP TO <tape name> ABORTED:  
STRUC FILE <file name> IS LOCKED OUT.

You have initiated DMSDLR and requested the DUMP option. A database structure file is locked out. Recover the file using reconstruct recovery and then try the dump again.

DATABASE ONLINE PARTIAL DUMP TO <tape name> ABORTED:  
STRUC FILE <file name> NOT RECOVERED FROM PREVIOUS ONLINE DUMP.

You have initiated DMSDLR and requested the DUMP option. If you load a structure from a backup tape that was created with an online dump, the structure must have a recovery procedure done on it before it can be dumped again. After you do the required recovery procedure, repeat the DUMP instruction.

DATABASE ONLINE PARTIAL DUMP TO <tape name> COMPLETED.

This is an information only message about the DUMP option of the DMSDLR utility. All specified files have been dumped.

DATABASE ONLINE PARTIAL DUMP TO <tape name> INITIATED.

You have initiated DMSDLR with the DUMP TO <tape name> option. This is an information only message.

DATA SET NAME OR SET NAME EXPECTED

You must enter a data set or set name after GENERATE.



#### DBP FAILED

CATEGORY = <category number>

SUBCATEGORY = <subcategory number>

STRUCTURE <structure number> OCCURRENCE <occurrence number>

This is a DBP error that applies to both Standard and Reorg DBPs. The DBP has encountered an irrecoverable error and has stopped. This chart shows certain category and subcategory numbers.

- **09 nn** - IOERROR - Take appropriate action to correct media error; then perform recovery.
- **07 31** - SYSTEMERROR(REBUILD REQUIRED) - A previous recovery action has failed. Perform rebuild recovery with a stopping point before the audit record on which it failed, or correct media problems.
- **07 32** - SYSTEMERROR(DATABASE NOT AUDITED) - You have made an attempt to recover an unaudited database. Load a prior dump and reprocess all transactions.
- **18 01** - VERSIONERROR(DBP-CONTROL FILE) - The DBP and control file version stamps do not match. Load the appropriate DBP or control file version (along with other database structure files).
- **18 04** - VERSIONERROR(DBP-MCP) - You have made an attempt to run a release version of the DBP on an MCP of a lower release version. Change to a release level of MCP equal to or greater than that of the DBP.

For all other errors take the same action as for FATALERROR. Refer to Section 8 for more information about FATALERROR.

#### DBP SOFT INTERRUPT AT <address>, <error type>

The DBP has encountered a processor error. This message applies to Standard DBPs only. The <error type> is one of the following:

SOFT ERROR <integer>  
STACK OVERFLOW  
DEREF NIL PTR  
DEREF UNINIT PTR  
ADDRESS ERROR  
INVALID ARITH DATA  
INVALID INSTRUCTION  
INSTRUCTION TIMEOUT  
MEMORY PARITY  
UNKNOWN ERROR

For SOFT ERROR the values of <integer> appear in the *B 2000/B 3000/B 4000 Series MCP Programmer's Guide* under the ARM BCT. For any of these errors, take the same action as you would for the FATALERROR message. Refer to Section 8 for more information about FATALERROR.

#### DISPLAY OPTION INVALID

You have used the SP command to display information about a database, but the characters you entered are not one of the keywords for the parameters that can be displayed (CONTROLPOINT, STACKSIZE and so on). Usually, this message indicates that you have entered a keyword incorrectly. Reenter the SP command with correct keywords and values.





#### DISPLAY OPTION MISSING

You have used the SP command to display database parameters, but you have not entered a parameter. Reenter the SP command with at least one of the valid display options (ALLOWEDCORE, STACKSIZE and so on).

#### DMS ERROR

This message identifies a DMS error on the operator display terminal (ODT). It is usually preceded by two asterisks (\*\*DMS ERROR) and is normally followed by one or more lines of specific error messages.

#### END COMPRESS <data set structure name>

This information only message comes from the Reorg DBP and indicates that the compression process is ending.

#### END COMPRESS <data set structure name> ORDERED BY <set structure name>

This information only message comes from the Reorg DBP. It indicates that the compression process with the ORDERED BY option is ending.

#### END COPY BACK <structure name>

This information only message comes from the Reorg DBP. It indicates that the COPY BACK function is ending.

#### END FIXUP <master data set structure name>

This information only message comes from the Reorg DBP.

#### END FIXUP <set/subset structure name>

This information only message comes from the Reorg DBP.

#### END GENERATE <set/subset structure name>

This information only message comes from the Reorg DBP and indicates that the generate process is ending.

#### END PREVIOUS RUN IR SYNTAX

DMSDLR displays this information only message to indicate that this is the end of the previous IR syntax.

#### ENDING <recovery type>

This is an information only message. The denoted recovery process has ended. For a list of recovery types, see BEGINNING <recovery type>.

#### ERROR OCCURRED IN PRODUCTION USER CODE

A DBP SORT INTERRUPT error has occurred in the Production DBP user code (that is, the host language code bound into the Production DBP.)



#### FATALERROR

SUBCATEGORY = <subcategory number for FATALERROR>

STRUCTURE <structure number> OCCURRENCE <occurrence number>

This is a DBP error that applies to both Standard and Reorg DBPs. The DBP has encountered an irrecoverable error and is terminating. Subcategory values of 1-18 and 61 are the same as the corresponding category. For all other values the DBP has detected a software error. The latter case should be reported to your Unisys representative along with a DBPDMP report, a bind listing (XXXBND) obtained by using the \$LIST BIND option, and any additional information that will help Unisys to solve the problem (for example, the DBP, database, and programs running at the time of failure, or some test case that causes the problem to occur). Refer to Section 8 for more information about FATALERROR.

#### FILE SWAPPING ERROR

The Reorg DBP creates a temporary file XX\*nnn when it generates a structure. After the generation is complete, the Reorg DBP removes the old structure and changes the name of the new file to that of the old structure. The Reorg DBP displays this message if it is unable to complete the swap.

#### FOLLOWING FILE NAME INVALID <file name>

The parameters for reconstruct recovery contained the designated structure file name and the DBP could not locate it in the database description. Check for misspellings. This process applies to a Standard DBP only.

#### FOLLOWING STRUC FILES WERE SUCCESSFULLY RECONSTRUCTED

See the message entitled "THE FOLLOWING STRUC FILES WERE SUCCESSFULLY RECONSTRUCTED."

#### FUNCTION NOT ALLOWED:

##### LOAD/AUTOLOAD CURRENTLY IN PROGRESS

This is an error message about the LOAD, REBUILD, PURGE, REORGANIZE, DUMP, RECONSTRUCT or ROLLBACK functions of DMSDLR utility that you initiated with the IR command or through the DMSUTL utility. Your request for a recovery function has been ignored because a database LOAD, REBUILD or RECONSTRUCT is currently in process. Repeat your request when this function finishes.

#### FUNCTION NOT ALLOWED - DO RESTART:

##### PREVIOUS LOAD/AUTOLOAD ABORTED

This is an error message about the LOAD, PURGE, REORGANIZE, DUMP, REBUILD, RECONSTRUCT or ROLLBACK functions of DMSDLR utility that you initiated with the IR command or through the DMSUTL utility. Your request for a recovery function has been ignored because a previous LOAD, REBUILD or RECONSTRUCT has failed to complete successfully. Recover with an IR RESTART command or through the RESTART option of the DMSUTL utility.

#### INTERNAL ALLOWEDCORE VALUE NOT CHANGED

##### CHANGE AFFECTS SUBSEQUENT RUNS ONLY

You have used the SP command to CHANGE the value of ALLOWEDCORE. However, the value would cause the database program to go from below 1MD to above 1MD or vice versa. The change will take place the next time the database program is initiated. The current value of ALLOWEDCORE remains unchanged.



#### INVALID COPY OPTION

The Reorg DBP requires information from the XXXRNF file. This message indicates that the XXXRNF file is corrupted and that the reorganization process is not restartable. You must remove the XXXRNF file and reenter the IR command.

#### INVALID REORG GLOBAL STATE

The Reorg DBP requires information from the XXXRNF file. This message indicates that the XXXRNF file is corrupted and that the reorganization process is not restartable. You must remove the XXXRNF file and reenter the IR command.

#### INVALID REORG STRUCTURE STATE

The Reorg DBP requires information from the XXXRNF file. This message indicates that the XXXRNF file is corrupted and that the reorganization process is not restartable. You must remove the XXXRNF file and reenter the IR command.

#### IR GENERATE SYNTAX RUN COMPLETED OK

This is an information only message from DMSREO about the GENERATE statement.

#### LOAD INITIATED ON DATABASE <database name>

This is an information only message from DMSDLR about offline dump.

#### LOADFACTOR OF <set name> = <integer>

You have used the SP command to display the current LOADFACTOR of <set name>.

#### LOADFACTOR OF <set name> CHANGED TO <integer>

You have used the SP command to change the LOADFACTOR of <set name> to the displayed value.

#### LOADFACTOR OF <set name> SET TO <integer>

You have used the SP command to set the LOADFACTOR of <set name> to the displayed value. This value will remain in effect for one execution of the DBP.

#### LOADFACTOR VALUE EXPECTED

You have used the SP command to change the LOADFACTOR value of a given set, but you have not entered a new LOADFACTOR value. Reenter the SP command with a LOADFACTOR value from 1 to 99.

#### LOADFACTOR VALUE MUST BE AN INTEGER

You have used the SP command to change the LOADFACTOR parameter of a given structure, but the characters you entered are not a valid number. Reenter the SP command with a valid number for LOADFACTOR.

#### LOADFACTOR VALUE OUT OF RANGE (1 - 99)

You have used the SP command to change the LOADFACTOR of a given set, but the number you entered is greater than 99 or less than 1. Reenter the SP command with a LOADFACTOR between 1 and 99.

#### MUST DO IR ON REORGANIZE DBP

You have made an attempt to issue an "IR <database name> REORGANIZE" command to the normal DBP. Reenter the command using the name of the Reorg DBP (nnnREO).



**NO AUDIT RECORDS FOUND FOR  
REBUILD RECOVERY**

The rebuild recovery process detected no updates that were not already in the database to be rebuilt.

**NO AUDIT RECORDS FOUND FOR  
RECONSTRUCT RECOVERY**

The reconstruct recovery process detected no files that are to be reconstructed.

**NO AUDIT RECORDS FOUND FOR  
ROLLBACK RECOVERY**

The rollback recovery process detected no files that are to be rolled-back.

**NO INPUT FOUND — SP COMMAND TERMINATED**

You have used the SP command to display or change parameters of a database, but the first word received with an AX command (ACCEPT) is END. Normal DBP processing resumes when this message is displayed. To enter more SP parameters, reenter the SP command and then use <mix no> AX commands to display or change parameters.

**NO MORE INPUT EXPECTED**

This is an information only message from DMSDLR about the reorganization process.

**NO RECORDS FOUND FOR REBUILD**

See the message entitled "ALREADY BEYOND SPECIFIED ABN."

**NO RECORDS FOUND FOR ROLLBACK**

See the message entitled "ALREADY AHEAD OF SPECIFIED ABN."

**NUMBER EXCEEDS 10 DIGITS**

You have used the SP command to enter a value that is more than 10 digits long. The SP command can only accept values less than or equal to 10 digits. Reenter the SP command with a smaller number.

**OFFLINE AUTOLOAD RECONSTRUCT INITIATED.**

This is an information only message.

**OFFLINE DUMP COMPLETED**

This is an information only message from DBP about offline dump.

**OFFLINE DBPDMP <tape name> OF DATABASE <database name> COMPLETE**

This is an information only message from DMSDLR about offline dump.

**OFFLINE DUMP IS IN PROGRESS, NEW OFFLINE DUMP HAS BEEN QUEUED**

This is an information only message from DMSDLR about offline dump.

**OFFLINE DUMP IS IN PROGRESS, NEW ONLINE DUMP HAS BEEN QUEUED**

This is an information only message from DMSDLR about offline dump.





OFFLINE DUMP IS IN PROGRESS, NEW UPDATER HAS BEEN QUEUED

This is an information only message from DMSDLR about offline dump.

OFFLINE DUMP INITIATED ON DATABASE <database name>

This is an information only message from DMSDLR about offline dump.

OFFLINE DUMP RUNNING

This is an information only message from DMSDLR about offline dump.

OFFLINE DUMP TERMINATING

This is an information only message from DMSDLR about offline dump.

OFFLINE PARTIAL DUMP INITIATED.

This is an information only message.

OLD CNF AHEAD OF NEW CNF

This error message is about reconstruct recovery. The old control file (nnnCNO), which was loaded from the archive tape, is determined by the timestamp to be not older than the current control file (nnnCNF). Be sure that the old control file was properly renamed after it was loaded.

OLD CNF LOCK SHOULD BE:  
NOT LOCKED OR RECONSTRUCT RUNNING

-or-

OLD CNF IN USE VALUE SHOULD BE  
NO RECOVERY OR ELSE  
DISCONTINUITY REQUIRED

For a reconstruct recovery, the old control file must have the designated state. Load a control file and the file(s) to be reconstructed from a prior database dump.

ONLINE AUTOLOAD RECONSTRUCT INITIATED.

This is an information only message.

ONLINE DBPDMP <tape name> OF DATABASE <database name> COMPLETED

This is an information only message from DMSDLR about online dump.

ONLINE DUMP COMPLETED

This is an information only message from DBP about online dump.

ONLINE DUMP INITIATED ON DATABASE <database name>

This is an information only message from DMSDLR about online dump.

ONLINE DUMP IS IN PROGRESS, NEW OFFLINE DUMP HAS BEEN QUEUED

This is an information only message from DMSDLR about online dump.

ONLINE DUMP IS IN PROGRESS, NEW ONLINE DUMP HAS BEEN QUEUED

This is an information only message from DMSDLR about online dump.



ONLINE DUMP REQUEST DENIED, DATABASE NOT AUDITED

This is an information only message from DMSDLR about online dump.

ONLINE DUMP RUNNING

This is an information only message from DMSDLR about online dump.

ONLINE DUMP TAPE NAME <tape name> <dump-date-time-stamp> INVALIDATED BY ROLLBACK

This is an information only message from DMSDLR about online dump.

ONLINE DUMP TERMINATING

This is an information only message from DMSDLR about online dump.

ONLINE PARTIAL DUMP INITIATED

This is an information only message.

ORDER BY STRUCTURE MUST BE A SET

The set you specified in the ORDER BY option must reference the data set that you generated with the ORDER BY option. It cannot be a subset or another data set. Refer to Section 3 for more information about the ORDER BY option.

PARAMETER FILE INVALID - RESTART NOT POSSIBLE

This is a nonfatal error message from DMSDLR about online dump.

PARTIAL DUMP ABORTED - INVALID SYNTAX:  
DUPLICATE OR NONEXISTENT LOGICAL STRUCTURE NAME:  
<structure name>

This is a nonfatal error message about the DUMP function that you initiated with the IR DUMP command or through the DUMP option of the DMSUTL utility. You have requested a partial dump with an incorrect structure list. One or more of the structures you requested is not present, or you named the same structure twice. Repeat the DUMP request with the correct structure list and database name.

PARTIAL DUMP ABORTED - INVALID SYNTAX:  
NONEXISTENT PHYSICAL STRUC NAME: <structure name>

This is a nonfatal error message about the DUMP function that you initiated with the IR DUMP command or through the DUMP option of the DMSUTL utility. You have requested a partial dump with an incorrect structure list. One or more of the structures you requested is not present, or you named the same structure twice. Repeat the DUMP request with the correct structure list and database name.

PLEASE INPUT ONE OF THE FOLLOWING:

SYN: DO A SYNTAX RUN AGAIN.

RES: RUN REORG OR GENERATE RESTART.

REM:FORGET PREV REORG OR GENERATE RUN.

REMOVE XXXRNF, DO NEW REORG OR GENERATE.

EXI: EXIT AND DO NOTHING.

When you do a RESTART function and whenever DMSDLR detects that there is an XXXRNF file on disk, this message will appear. The point at which the previous reorganization was stopped determines if the SYN option appears in the message.



PLEASE INPUT ONE OF THE FOLLOWING:

RES: RUN REORG OR GENERATE RESTART.

REM: FORGET PREV REORG OR GENERATE RUN.

REMOVE XXXRNF AND EXIT.

RGO: FORGET PREV REORG OR GENERATE RUN.

REMOVE XXXRNF, DO NEW REORG OR GENERATE.

EXI: EXIT AND DO NOTHING.

When you do a REORGANIZE function and whenever DMSDLR detects that there is an XXXRNF file on disk, this message will appear.

PLEASE RELOAD THE FOLLOWING FILES:

<structure list>

END OF RELOAD FILES.

AX AFTER THESE FILES ARE RELOADED.

This message indicates that REORGANIZATION RESTART has been invoked and DMSDLR has determined that certain files must be reloaded.

PREV ONLINE RECONSTRUCT COMPLETED OK

This is an information only message.

PREVIOUS COMMAND ENTERED IS: "<previous command syntax>"

IS THIS TO BE RESTARTED? YES OR NO

A LOAD, DUMP, or RECOVERY function failed, and you requested the RESTART function. The DBP has checked the file on disk that passes syntax to the RESTART function, and has determined that the named function is the one that is to be restarted. If you enter YES, the function will resume from the point of failure. One reason to enter NO would be if the wrong pass file was on disk and the stated function was not really the one that should be restarted.

PURGE FAILED; HALT/LOAD REQUIRED

The DBP displays this message during a PURGE function. The PURGE function has failed. If a halt/load is done, the DBP will finish the PURGE function and leave the database in a stable, accurate state. Refer to Section 3 for more information about the PURGE function.

PURGE RECORD FOUND IN FILE <file name>

This is an information only message that appears during the rebuild recovery process. Rebuild recovery is operating normally, but has encountered a PURGE record in the audit trail. File name is the name of the audit file being read. You do not need to do anything.

PURGE RECORD FOUND. ROLLBACK TERMINATING

Rollback recovery cannot extend backward in time over the audit trail discontinuity that the PURGE function creates. Refer to Section 3 for more information about the PURGE function.

READ ERROR— ADDR CHECK FAILED

STRUCTURE <structure number> BLOCK <block number>

RETRY FAILED

This message indicates a DBP error. It applies to the Standard DBP only. The DBP has detected an ADDRESSCHECK error during I/O (Input/Output) to the indicated structure and block. The DBP has retried the I/O operation and failed. This message indicates that there is a problem with the storage media. Correct the media problems before you continue to use the database.



READ ERROR— ADDR CHECK FAILED  
STRUCTURE <structure number> BLOCK <block number>  
RETRY SUCCESSFUL

This message indicates a DBP error. It applies to the Standard DBP only. The DBP has detected an ADDRESSCHECK error during I/O to the indicated structure and block. The DBP has retried the I/O operation and succeeded on the retry. You should correct the media problems before you continue to use the database.

READ ERROR— CHECKSUM FAILED  
STRUCTURE <structure number> BLOCK <block number>  
RETRY FAILED

This message indicates a DBP error. It applies to the Standard DBP only. The DBP has detected a CHECKSUM error during I/O to the indicated structure and block. The DBP has retried the I/O operation and failed. You should correct the media problems before you continue to use the database.

READ ERROR— CHECKSUM FAILED  
STRUCTURE <structure number> BLOCK <block number>  
RETRY SUCCESSFUL

This message indicates a DBP error. It applies to the Standard DBP only. The DBP has detected a CHECKSUM error during I/O to the indicated structure and block. The DBP has retried the I/O operation and succeeded. This message indicates that there are problems with the storage media. You should correct the media problems before you continue to use the database.

REBUILD FAILED ON PURGE RECORD. RESTART REBUILD

This message indicates that a rebuild recovery has failed when a PURGE record was encountered in the audit trail. To correct the failure, reenter the IR REBUILD command. Rebuild recovery will resume from the point of failure. Refer to Section 3 for more information about the PURGE function.

REBUILD/RECONSTRUCT REQUIRED

A recovery process has failed and you must recover the database using a previous backup of the database. If the recovery failed because of an error in the audit trail, then you should specify rebuild recovery with a stopping point before the audit trail block or file in error, and reprocess lost transactions with application programs. This process applies to a Standard DBP only.

REBUILD RECOVERY FAILED, START REBUILD AGAIN

A rebuild recovery failed when the recovery process encountered a PURGE record in the audit trail. To correct this, repeat rebuild recovery by reloading the backup and repeating the IR REBUILD command. Refer to Section 3 for more information about the PURGE function.

REBUILT THRU FILE <audit file name>

The DBP has completed the processing through the designated audit file for rebuild recovery. This process applies to a Standard DBP only.

RECONSTRUCT FAILED ON PURGE RECORD. RESTART RECONSTRUCT.

A reconstruct recovery failed when the recovery process encountered a PURGE record in the audit trail. To correct this, reenter the IR RECONSTRUCT <reconstruct list> command or repeat the RECONSTRUCT option through the DMSUTL utility. Reconstruct recovery will resume from the point of failure.





## RECOVERY ERROR

SUBCATEGORY = <subcategory number for RECOVERY ERROR>

STRUCTURE <structure number> OCCURRENCE <occurrence number>

This message indicates a DBP error. It applies only to Standard DBPs. The DBP has encountered an irrecoverable error during a recovery process and is terminating. Take the same action as for FATALERROR (include the audit trail in the information supplied); then perform a rebuild recovery to a point in the audit trail before the record on which it failed. Refer to Section 8 for more information about FATALERROR.

## RECOVERY FAILED IN FILE <audit file name>

A recovery process has failed while it was processing the indicated audit file. Correct any media problems. Retry the recovery or load a previous backup of the database and perform a rebuild recovery with a stopping point before the audit file in which the error occurred.

## RECOVERY IN PROCESS OR PENDING

SP COMMAND NOT ALLOWED

ANY QUEUED SP INPUT WILL BE

PROCESSED BY THE NEXT SP COMMAND

This message appears on your terminal if an online reconstruct recovery or abort recovery is pending. Any SP command input that you have entered remains in the queue.

## RECOVERY NOT REQUIRED

You have requested a halt/load or rollback recovery for a newly created database that did not require recovery.

## REGENERATE SUBSET IS NOT ALLOWED BY DMSREO

You cannot generate a new subset from DMSDLR. Use DASDL reorganization GENERATE.

## REORGANIZATION RUNNING

This information only message comes from the Reorg DBP.

## REORGANIZE STRUCTURE MUST BE A PHYSICAL STRUCTURE

You can reorganize only physical structures.

REORGANIZE ON <database name> ABORTED:

DATABASE IS IN USE

You have requested a reorganization of a database that is in use. DMSDLR checks the control file and has found that the database is already locked.

REORGANIZE ON <database name> ABORTED:

DATA SET NAME OR SET NAME IS REQUIRED - <name>

DMSDLR displays this message when you have requested a reorganization of a structure that does not exist in the database requested.

REORGANIZE ON <database name> ABORTED:

DUPLICATE GENERATE ON THE SAME STRUCTURE - <structure name>

DMSDLR displays this message when you have requested a reorganization of a structure more than once in the same IR command.

REORGANIZE ON <database name> ABORTED:

GENERATE OR RESTART KEYWORD EXPECTED - <id>

You have requested a reorganization of a database but have not specified either a GENERATE or RESTART keyword in the command.



REORGANIZE ON <database name> ABORTED:  
GENERATE STRUCTURE COUNT EXCEEDED LIMIT - <num>

You have attempted to reorganize more structures than are allowed for one reorganization. DMSDLR displays this message to indicate the maximum number of structures that can be reorganized with a single reorganization.

REORGANIZE ON <database name> ABORTED:  
INVALID MEDIUM NAME - <familyname>

DMSDLR displays this message to indicate that the medium name you have entered is invalid.

REORGANIZE ON <database name> ABORTED:  
INVALID REORG CONTROL FILE - <file id>

DMSDLR displays this message to indicate that the XXXRNF file is invalid.

REORGANIZE ON <database name> ABORTED:  
MEDIUM NAME EXPECTED

DMSDLR expected you to enter a medium name.

REORGANIZE ON <database name> ABORTED:  
MISSING KEYWORD - <key word>

DMSDLR expected you to enter a keyword that is not present.

REORGANIZE ON <database name> ABORTED:  
ONLY SET OR SUBSET CAN BE BALANCED - <name>

DMSDLR displays this message when the structure to be balanced is not a set or subset.

REORGANIZE ON <database name> ABORTED:  
ONLY STANDARD/DISJOINT UNORDERED MAY USE ORDER BY OPTION

DMSDLR displays this message to indicate that the structure using the ORDER BY option is not a disjoint standard or unordered data set.

REORGANIZE ON <database name> ABORTED:  
ORDER BY STRUCTURE IS NOT A PHYSICAL STRUCTURE - <set name>

DMSDLR displays this message to indicate that the structure to be used to order the data set is not a physical structure.

REORGANIZE ON <database name> ABORTED:  
ORDER BY STRUCTURE MUST BE A SET - <set name>

DMSDLR displays this message to indicate that the structure to be used to order the data set is not a set but is a subset. Only sets can be used to order a data set. Automatic and manual subsets cannot be used.

REORGANIZE ON <database name> ABORTED:  
REGENERATE EMBEDDED SET NOT ALLOWED BY DMSREO - <embedded set>

DMSDLR displays this message to indicate that the structure specified in the input parameters is an embedded structure. DMSREO does not allow generation of an embedded structure.

REORGANIZE ON <database name> ABORTED:  
REGENERATE SUBSET NOT ALLOWED BY DMSREO - <subset name>

DMSDLR displays this message to indicate that the structure specified in the input parameters is a subset. DMSREO does not allow generation of a subset.



REORGANIZE ON <database name> ABORTED:  
RELOAD FILES EXCEEDED LIMIT - <num>

DMSDLR displays this message to indicate that the number of files to be reloaded exceeds the limit.

REORGANIZE ON <database name> ABORTED:  
SET DOESN'T REFERENCE DATA SET  
SET: <set name>  
DATA SET: <ds name>

DMSDLR displays this message to indicate that the set specified does not reference the data set specified.

REORGANIZE ON <database name> ABORTED:  
STRUCTURE NAME EXPECTED - <name>

DMSDLR displays this message to indicate that the structure name specified is invalid.

REORGANIZE ON <database name> ABORTED:  
STRUCTURE WITH ORDER BY OPTION MUST BE A DATA SET - <name>

DMSDLR displays this message to indicate that the structure to be ordered is not a data set.

REORGANIZE ON <database name> ABORTED:  
SYNTAX KEYWORD EXPECTED - <id>

DMSDLR expected the keyword SYNTAX.

REORGANIZE ON <database name> ABORTED:  
THIS STRUCTURE IS NOT A PHYSICAL STRUCTURE - <name>

DMSDLR displays this message when the structure specified is not a physical structure.

REORGANIZE RESTART IN PROGRESS

DMSDLR displays this information only message to confirm that the reorganization restart process is beginning.

REQUIRED FILE —  
PROCESSING A RECOVERY RECORD

-or-

REQUIRED FILE FOR  
RECOVERY ROLLBACK

Rebuild or rollback recovery is processing a recovery audit record that requires the file not be processed with the OF command. You must load the audit file required by the DBP. This process applies to a Standard DBP only.

REQUIRED ITEM NULL  
STRUCTURE = <structure name>  
RRN = <integer>

You have added the REQUIRED option to a data set or item, and the Reorg DBP has encountered an item with a null value.

RESTART FROM THE BEGINNING

You have requested a REORGANIZATION RESTART. The Reorg DBP displays this message to indicate where the reorganization will restart.



**RESTART NOT ALLOWED:  
LOAD/AUTOLOAD CURRENTLY IN PROGRESS**

This error message is about the RESTART function of the DMSDLR utility that you initiated with the IR RESTART command or through the RESTART option of the DMSUTL utility. Your RESTART request has been ignored because there is a LOAD function or rebuild or reconstruct recovery currently in process. Because this function is currently running, you do not need to do a RESTART.

**RESTART NOT ALLOWED:  
NO PREVIOUS ABORTED LOAD/AUTOLOAD KNOWN**

This error message is about the RESTART function of DMSDLR utility that you initiated with the IR RESTART command or through the RESTART option of the DMSUTL utility. Your RESTART request has been ignored because the database has no memory of a previous failed LOAD function or rebuild or reconstruct recovery. A previous RESTART command where the 'REM' option was used could cause this. Reenter the desired recovery command.

**RESTARTING AT <structure name>**

You have requested a REORGANIZATION RESTART. The Reorg DBP displays this message to indicate where the reorganization will restart.

**RNF VERSION ERROR**

The version date-timestamp in the memory file does not match the date-timestamp in the XXXRNF file.

**ROLLBACK TERMINATING AT PURGE RECORD**

AUDITFILE = <file name>

AUDITBLOCK = <audit block number>

A rollback recovery cannot cross the audit trail discontinuity that a PURGE created. The rollback recovery encountered a PURGE record in the audit trail and terminated. The database is left in a stable, usable state. The file name is the name of audit file in which the PURGE discontinuity was detected. The audit block number is the number of audit block in which the discontinuity was detected. Refer to Section 3 for more information about the PURGE function.

**ROLLBACK TO FILE <audit file name>**

The DBP has completed processing through the designated audit file for rollback recovery. This process applies to a Standard DBP only.

**SET COMMAND INVALID FOR CLOSED STRUCTURE**

You have used the SP command to set a parameter for one DBP execution, but the user program has not invoked the structure involved. The parameters of a closed structure cannot be set for one execution of the DBP. Reenter SP command with a different structure name or change the user program to invoke the structure when the database is opened.

**SET DOES NOT REFERENCE THE DATA SET**

The set you specified in the ORDER BY option must reference the data set that you generated with the ORDER BY option. Refer to Section 3 for more information about the ORDER BY option.





#### SET OPTION INVALID

You have used the SP command to set a database parameter for one DBP execution, but the characters you entered are not one of the valid parameters (STACKSIZE, LOADFACTOR and so on). Usually, this message indicates that you have entered the keyword for a parameter incorrectly. Reenter the SP command with valid keywords and values.

#### SET OPTION MISSING

You have used the SP command to set the value of a database parameter for one DBP execution, but you have not entered a parameter. Reenter the SP command with at least one valid option (LOADFACTOR, WRITEAHEAD and so on).

#### SP COMMAND PROCESSING COMPLETE FOR <database name> NORMAL DBP PROCESSING RESUMED

This message indicates that the word END has been entered through a <mix no> AX command to terminate SP command processing. When this message appears, normal database operations resume. To change other parameters, enter another SP command.

#### SP COMMAND PROCESSING INITIATED FOR <database name> NORMAL DBP PROCESSING INTERRUPTED DMS/<database name> = <mix no> ENTER SP PARAMETERS

This message is the first response to an SP command and indicates that the DBP is ready to receive additional commands through <mix no> AX commands.

#### STACKSIZE = <integer> DIGITS

You have used the SP command to display the current STACKSIZE value.

#### STACKSIZE CHANGED TO <integer> DIGITS

You have used the SP command to change STACKSIZE to the displayed value.

#### STACKSIZE SET TO <integer> DIGITS

You have used the SP command to set STACKSIZE to the displayed value. This value will remain in effect for one execution of the DBP.

#### STACKSIZE VALUE CANNOT EXCEED <integer>

The number you entered for STACKSIZE with the SP command is greater than the maximum limit. When you are running under MCP/IX, this limit is six million digits. When you run under MCP/VS 1.0 and 2.0, the limit is 5,990,000 digits. Reenter the SP command with a smaller number for STACKSIZE.

#### STACKSIZE VALUE EXCEEDS ALLOWEDCORE

The number you entered for STACKSIZE with the SP command is greater than the current ALLOWEDCORE value. Enter a smaller number for STACKSIZE or change ALLOWEDCORE to a larger number.

#### STACKSIZE VALUE EXPECTED

You have used the SP command to change the value of STACKSIZE, but you have not entered a new STACKSIZE value. Reenter the SP command with a valid number for the STACKSIZE value.



#### STACKSIZE VALUE MUST BE AN INTEGER

You have used the SP command to change the STACKSIZE value, but the characters you entered as the new value are not a valid number. Reenter the SP command with a valid number for STACKSIZE.

#### STACKSIZE VALUE MUST BE GREATER THAN <integer>

The number you entered for STACKSIZE with the SP command is less than the minimum limit. The DASDL compiler sets the minimum limit and it varies depending on the \$BIND compiler control option, and other DASDL options. Reenter the SP command with a larger number for STACKSIZE.

#### STATISTICS OPTION SET

You have used the SP command to display the status of the STATISTICS option.

#### STRUCTURE ID EXPECTED

You have used the SP command to change or to display one of the parameters (BUFFERS, WRITEAHEAD and so forth) of a given structure, but you have not entered a structure name. Reenter the SP command with a valid structure name.

#### STRUCTURE ID NOT FOUND IN MEM FILE

You have used the SP command to change a parameter of a given structure, but the name of that structure does not reside in the current memory file of the database. This message indicates that the memory file may be corrupt. Check to be certain that the memory file has not been inadvertently replaced or damaged.

#### STRUCTURE MUST BE EITHER A SET OR SUBSET

You have used the SP command to change, set or display a parameter (for example, LOADFACTOR) of a given set, but the name you entered is not the name of a set or subset. Reenter the SP command with a valid set or subset name.

#### STRUCTURE NOT IN DATABASE

You have used the SP command to enter a <structure name>, but the structure you entered does not exist in the structure parameters of the current database. Usually, this message indicates that you have entered the name of the structure incorrectly. Reenter the SP command with a valid structure name.

#### SWITCH KEYWORD EXPECTED — AUDITFILE

You have used the SP command to close the current audit trail files and open the next sequential audit files, but you have omitted the keyword AUDITFILE. Reenter the SP command with the keyword AUDITFILE.

#### SWITCH OPTION INVALID

You have used the SP command to close a database's current audit trail files and open the next sequential audit trail files, but you have omitted the keyword AUDITFILE. Reenter the SP command including the keyword AUDITFILE.

#### SYNCPOINT = <integer> TRANSACTIONS

You have used the SP command to display the current SYNCPOINT value.



#### SYNCPOINT CHANGED TO <integer> TRANSACTIONS

You have used the SP command to change the SYNCPOINT to the displayed value.

#### SYNCPOINT SET TO <integer> TRANSACTIONS

You have used the SP command to set SYNCPOINT to the displayed value. This value will remain in effect for one execution of the DBP.

#### SYNCPOINT VALUE EXPECTED

You have used the SP command to change the value of SYNCPOINT, but you have not entered a new SYNCPOINT value. Reenter the SP command with a valid number for the SYNCPOINT value.

#### SYNCPOINT VALUE MUST BE AN INTEGER

You have used the SP command to change the SYNCPOINT value, but the characters you entered as the new value are not a valid number. Reenter the SP command with a valid number for SYNCPOINT.

#### SYNCPOINT VALUE OUT OF RANGE (1 - 9999)

The SYNCPOINT value you entered with the SP command is outside of the allowed range. The number of transactions per syncpoint cannot be less than 1 or greater than 9999. Reenter the SP command with a valid value for SYNCPOINT.

#### THE FOLLOWING FILES MUST HAVE BEEN RELOADED

When a reorganization is restarted, DMSREO or the Reorg DBP analyzes the current state and determines the restart point. Some files may have to be reloaded from the dump taken before the reorganization. DMSDLR identifies this file list and you must reload these files. This message is a reminder that you should have reloaded these files.

#### THE FOLLOWING STRUC FILE(S) ARE CURRENTLY LOCKED

OUT:<structure names>

#### THESE FILES SHOULD BE RECONSTRUCTED AS SOON AS POSSIBLE!!

This message is a summary of all locked out structures in the database that appears when the DBP performs an open, close, or haltload recovery. A structure becomes locked out when it is unstable and in need of recovery. You can solve the problem with a reconstruct recovery. Reconstruct recovery loads a stable version of the structure from a backup tape. Refer to Section 5 for more information about how to dump and reload a database. Any updates to these structures that a host language program tries to do will be denied and it will receive a structure locked out error (9, 15).

#### THE FOLLOWING STRUC FILES WERE

#### SUCCESSFULLY RECONSTRUCTED

THRU AUDIT FILE <audit file name>

<structure name list>

This message indicates that the DBP has completed the processing through the designated audit file for reconstruct recovery of the listed files.



THE PREVIOUS ABORTED FUNCTION WAS:

<IR syntax that initiated the aborted function>

PLEASE INPUT ONE OF THE FOLLOWING:

RES — RESTART THE FAILED FUNCTION.

EXI — EXIT AND DO NOTHING.

REM— REMOVE THE MEMORY OF THE FAILED FUNCTION  
(DOES NOT BACK OUT THE FAILED FUNCTION).

This message occurs in response to an IR RESTART command. The RESTART command itself is entered to resolve a failed LOAD, RECONSTRUCT or REBUILD function. (Note that most other IR commands are disallowed until the failed function is resolved.) To restart the failed function, enter <mix number> AX RES where the mix number is that of DMSDLR. If you are sure that the failed function can be ignored (for example you tried a reconstruct recovery which failed and now you want to do a rebuild recovery), enter <mix number> AX REM to remove the memory of the failed function from the database. Please note that this option does not back out or otherwise resolve the failed function! If you want to postpone any decisions about the failed function or are unsure what to do, enter <mix number> AX EXI to exit the RESTART function.

THE STRUCTURE WITH ORDER BY MUST BE A DATA SET

If you generated the structure with the ORDER BY option, the structure must be a data set. Refer to Section 3 for more information about the ORDER BY option.

TOKEN EXCEEDS 17 CHARACTERS

You have used the SP command to enter a name or keyword, but the entry is longer than 17 characters. Names or keywords you enter with the SP command must be less than or equal to 17 characters in length. Reenter the SP command with a shorter name or keyword.

UPDATER IS IN PROGRESS, NEW OFFLINE DUMP HAS BEEN QUEUED

This is an information only message about online dump.

USER TABLE OVERFLOW

DS OR DP

You have specified with the LIMIT DMSDBP and LIMIT DMSUSERS records the number of database programs allowed in the mix at any one time. These LIMIT records control the size of specific internal DMS tables. This message indicates that too many programs are entering the mix. You must terminate the program that caused the overflow (that is, the program whose mix number is associated with this message).

VERIFY CONDITION FAILED

STRUCTURE = <structure name>

RRN = <integer>

You have added or changed the VERIFY option in the DASDL source and the Reorg DBP has encountered a database record that does not satisfy the new condition.

WARNING: ONLY FULL DATABASE DUMP TAPES  
MAY BE USED WITH AUTOLOAD REBUILD.

This is an information only message that appears when a rebuild recovery begins.

WARNING: ONLY FULL OFFLINE DATABASE DUMP TAPES  
MAY BE USED WITH MANUAL LOAD.

This is an information only message that appears when a database LOAD starts.





#### WRITEAHEAD BOOLEAN EXPECTED

You have used the SP command to change the WRITEAHEAD option of a given structure, but you have entered neither TRUE nor FALSE. Reenter the SP command and set WRITEAHEAD to either TRUE or FALSE (WRITEAHEAD FOR <structure name> = <boolean>).

#### WRITEAHEAD FOR <structure name> = <boolean>

You have used the SP command to display the current status of the WRITEAHEAD option.

#### WRITEAHEAD FOR <structure name> CHANGED TO <boolean>

You have used the SP command to change the WRITEAHEAD option of <structure name> to the displayed value.

#### WRITEAHEAD FOR <structure name> SET TO <boolean>

You have used the SP command to set the WRITEAHEAD option of <structure name> to the displayed value. This value will remain in effect for one execution of the DBP.

#### WRITEAHEAD VALUE MUST BE EITHER TRUE OR FALSE

You have used the SP command to change the WRITEAHEAD option of a structure, but the choice you entered is other than TRUE or FALSE. Reenter either TRUE or FALSE with the SP command.

#### WRONG RNF FILE FOR REORGANIZATION

The Reorganization Information File (XXXRNF) used for a REORGANIZATION RESTART and IR GENERATE does not correspond to the current reorganization request. The file may be left over from a previous reorganization.



## SECTION 8

### EXCEPTION CONDITIONS FOR USER PROGRAMS

#### INTRODUCTION

Table 8-1 contains a list by category and subcategory number of exceptions that can occur during the execution of a database program (DBP) in the Data Management System II (DMSII). The DBP places a four-digit number, the concatenation of the category number and the subcategory number, in the user program's DMSTATUS register when an exception occurs.

Table 8-1. Exception Categories

| Category Number | Sub-Category Name | Number | Description                                                               |
|-----------------|-------------------|--------|---------------------------------------------------------------------------|
| 00              | SUCCESSFUL        | 00     | Function successful, no errors                                            |
| 01              | NOTFOUND          | 01     | No record satisfying condition                                            |
|                 |                   | 02     | No more records in next direction                                         |
|                 |                   | 03     | No more records in prior direction                                        |
|                 |                   | 04     | Record not in manual subset                                               |
|                 |                   | 05     | Embedded member does not exist                                            |
|                 |                   | 06     | Record not in auto subset                                                 |
| 02              | DUPLICATES        | 01     | Duplicates not allowed (direct data set)                                  |
|                 |                   | 02     | Duplicates not allowed (set or subset)                                    |
| 03              | DEADLOCK          | 01     | Deadly embrace condition was found while trying to MODIFY a record        |
|                 |                   | 02     | Program already has performed a MODIFY on the record in another work area |
|                 |                   | 03     | Deadlock time-out while trying to MODIFY a record                         |



**Table 8-1. Exception Categories (Cont.)**

| <b>Category Number</b> | <b>Sub-Category Name</b> | <b>Number</b> | <b>Description</b>                                                                        |
|------------------------|--------------------------|---------------|-------------------------------------------------------------------------------------------|
| 03                     | DEADLOCK                 | 04            | Deadlock detected while awaiting syncpoint                                                |
| 04                     | DATAERROR                | 01            | A REQUIRED item is null (including KEY items).                                            |
|                        |                          | 02            | A READONLY item was changed (remap).                                                      |
|                        |                          | 03            | The VERIFY condition not satisfied (data set or remap).                                   |
|                        |                          | 04            | A direct data set record key field contains zero or undigits                              |
| 05                     | NOTLOCKED                | 01            | STORE operation not preceded by CREATE, MODIFY, or STORE                                  |
|                        |                          | 02            | Attempt to update detail when master not locked                                           |
| 06                     | KEYCHANGED               | 01            | An attempt has been made to change the key value of a random data set on MODIFY-STORE.    |
|                        |                          | 02            | An attempt has been made to change a key value in a set where DUPLICATES are not allowed. |
| 07                     | SYSTEMERROR              | 10            | Attempt to access a structure which was not declared                                      |
|                        |                          | 20            | Structure not found                                                                       |
|                        |                          | 21            | Current record or path not found                                                          |
|                        |                          | 22            | Referenced structure not found                                                            |
|                        |                          | 23            | Embedded structure not found                                                              |



**Table 8-1. Exception Categories (Cont.)**

| <b>Category<br/>Number</b> | <b>Sub-Category<br/>Name</b> | <b>Number</b> | <b>Description</b>                                             |
|----------------------------|------------------------------|---------------|----------------------------------------------------------------|
| 07                         | SYSTEMERROR                  | 29            | Recovery request rejected because database dump is in progress |
|                            |                              | 30            | Recovery not required                                          |
|                            |                              | 31            | Rebuild of database required                                   |
|                            |                              | 32            | Recovery or online dump of unaudited database attempted        |
|                            |                              | 33            | Array of structures to reconstruct is invalid                  |
|                            |                              | 34            | Duplicated audit not specified                                 |
|                            |                              | 35            | Halt/load required                                             |
|                            |                              | 36            | Reconstruct recovery required                                  |
|                            |                              | 38            | File returned from an online dump needs recovery               |
|                            |                              | 39            | Structure returned from an online dump needs recovery          |
|                            |                              | 40            | Not a Production DBP                                           |
|                            |                              | 50            | Non-implemented organization type                              |
|                            |                              | 51            | Structure restriction fault                                    |
|                            |                              | 70            | Invalid function operation                                     |
|                            |                              | 71            | Invalid function parameter                                     |
|                            |                              | 72            | Invalid Structure Information Block                            |
|                            |                              | 73            | Invalid set selection expression                               |
|                            |                              | 74            | Database in use                                                |





B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Exception Conditions for User Programs

---

**Table 8-1. Exception Categories (Cont.)**

| Category Number | Sub-Category Name | Number | Description                                                                                |
|-----------------|-------------------|--------|--------------------------------------------------------------------------------------------|
| 07              | SYSTEMERROR       | 80     | MCP data transfer failure                                                                  |
|                 |                   | 91     | Wrong function order                                                                       |
|                 |                   | 92     | Function not allowed                                                                       |
| 08              | READONLY          | 01     | Attempt to change database when opened INQUIRY                                             |
| 09              | IOERROR           | 01     | I/O unit not ready                                                                         |
|                 |                   | 02     | Read parity error                                                                          |
|                 |                   | 04     | I/O descriptor error                                                                       |
|                 |                   | 06     | I/O End of File                                                                            |
|                 |                   | 09     | I/O MCP error                                                                              |
|                 |                   | 10     | I/O CHECKSUM error                                                                         |
|                 |                   | 11     | I/O address check error                                                                    |
|                 |                   | 15     | I/O lockout error - structure is locked out                                                |
| 10              | LIMITERROR        | 20     | I/O file boundary error                                                                    |
|                 |                   | 02     | Maximum number of users exceeded<br>1. MCP tables exceeded<br>2. No stack space in the DBP |
|                 |                   | 03     | Attempt to store too many records in a structure                                           |
|                 |                   | 04     | REORGANIZE structure overflow                                                              |
|                 |                   | 05     | REORG SORT/MERGE file size error                                                           |



**Table 8-1. Exception Categories (Cont.)**

| <b>Category<br/>Number</b> | <b>Sub-Category<br/>Name</b> | <b>Number</b> | <b>Description</b>                                                                     |
|----------------------------|------------------------------|---------------|----------------------------------------------------------------------------------------|
| 11                         | OPENERERROR                  | 01            | MCP open error                                                                         |
|                            |                              | 02            | Database program not on disk                                                           |
|                            |                              | 05            | Attempt to open a structure which has been locked out due to structure inconsistencies |
|                            |                              | 06            | Attempt to open database when it is already open                                       |
|                            |                              | 07            | Attempt to open a database which is closing                                            |
|                            |                              | 08            | Recovery in process                                                                    |
|                            |                              | 10            | Rebuild recovery required                                                              |
|                            |                              | 11            | Lockout of database files failed                                                       |
|                            |                              | 12            | Conversion of database to current release required                                     |
|                            |                              | 13            | Reorganization required                                                                |
|                            |                              | 14            | Deleted structure invoked                                                              |
|                            |                              | 15            | PRODUCTION user running                                                                |
|                            |                              | 16            | Online dump attempted for an unaudited database                                        |
|                            |                              | 20            | Control file locked by reorganization                                                  |
|                            |                              | 21            | Control file locked by halt/load recovery                                              |
|                            |                              | 22            | Control file locked by rebuild recovery                                                |
|                            |                              | 23            | Control file locked by reconstruct recovery                                            |



**Table 8-1. Exception Categories (Cont.)**

| Category Number | Sub-Category Name | Number | Description                                                                                   |
|-----------------|-------------------|--------|-----------------------------------------------------------------------------------------------|
| 11              | OPENERERROR       | 24     | Control file locked by rollback recovery                                                      |
|                 |                   | 25     | Control file locked by DMSTAT                                                                 |
|                 |                   | 26     | Control file locked by CNV640                                                                 |
|                 |                   | 27     | Control file locked by DMSDAP                                                                 |
|                 |                   | 28     | Control file locked by DASDL                                                                  |
| 12              | CLOSEERROR        | 29 01  | Control file locked by PURGE<br>Error in MCP while closing file                               |
|                 |                   | 10     | CLOSE of database while in transaction state                                                  |
| 13              | NORECORD          | 01     | Master current record not defined                                                             |
|                 |                   | 02     | Current record not defined                                                                    |
|                 |                   | 03     | Current path not defined                                                                      |
|                 |                   | 04     | Master of a subset and the master of a data set record being inserted are not in same subfile |
| 14              | INUSE             | 03     | Attempt to DELETE master record when embedded structure has members                           |
| 15              | AUDITERERROR      | 01     | Already in transaction state                                                                  |
|                 |                   | 02     | Already out of transaction state                                                              |
|                 |                   | 03     | Update attempted while not in transaction state                                               |
|                 |                   | 04     | Write error on Block one of audit trail                                                       |



**Table 8-1. Exception Categories (Cont.)**

| Category Number | Sub-Category Name | Number | Description                                |
|-----------------|-------------------|--------|--------------------------------------------|
| 15              | AUDITERROR        | 05     | CHECKSUM error on Block one of audit trail |
|                 |                   | 06     | Audit file mismatch                        |
| 16              | ABORT             | 01     | Transaction rolled back                    |
|                 |                   | 02     | Abort on BEGIN-TRANSACTION                 |
|                 |                   | 03     | Abort on END-TRANSACTION                   |
|                 |                   | 04     | Abort on CLOSE                             |
| 18              | VERSIONERROR      | 01     | DBP control file mismatch                  |
|                 |                   | 02     | Host program file mismatch                 |
|                 |                   | 03     | Update level mismatch                      |
|                 |                   | 04     | MCP release level mismatch                 |
|                 |                   | 05     | Memory file mismatch                       |
|                 |                   | 06     | Host program-MCP release level mismatch    |
| 19              | FATALERROR        | 22     | Open-close list link error                 |
|                 |                   | 23     | Lock list link error                       |
|                 |                   | 24     | Block list link error                      |
|                 |                   | 25     | Block lock list link error                 |
|                 |                   | 26     | Free list link error                       |
|                 |                   | 27     | Restart list link error                    |
|                 |                   | 28     | Task list link error                       |
|                 |                   | 29     | Dynamic memory link error                  |
|                 |                   | 30     | Sync list link error                       |





B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Exception Conditions for User Programs

---

**Table 8-1. Exception Categories (Cont.)**

| <b>Category<br/>Number</b> | <b>Sub-Category<br/>Name</b> | <b>Number</b> | <b>Description</b>                               |
|----------------------------|------------------------------|---------------|--------------------------------------------------|
| 19                         | FATALERROR                   | 31            | Active list link error                           |
|                            |                              | 32            | OPEN-CLOSE request error                         |
|                            |                              | 33            | Lock request error                               |
|                            |                              | 34            | Block request error                              |
|                            |                              | 35            | Block lock request error                         |
|                            |                              | 36            | Free list request error                          |
|                            |                              | 37            | Restart request error                            |
|                            |                              | 38            | Locked block error                               |
|                            |                              | 39            | Dynamic memory request error                     |
|                            |                              | 40            | Database parameter field<br>invalid              |
|                            |                              | 41            | Task field invalid                               |
|                            |                              | 42            | Reference to embedded structure<br>field invalid |
|                            |                              | 43            | Block control structure field<br>invalid         |
|                            |                              | 44            | Key descriptor field invalid                     |
|                            |                              | 45            | Current record or path field<br>invalid          |
|                            |                              | 46            | Structure field invalid                          |
|                            |                              | 47            | Dump list link error                             |
|                            |                              | 48            | Open wait dump list link error                   |
|                            |                              | 49            | Dump wait controlpoint flush<br>list link error  |
|                            |                              | 50            | Control file not found                           |



Table 8-1. Exception Categories (Cont.)

| Category Number | Sub-Category Name | Number | Description                                      |
|-----------------|-------------------|--------|--------------------------------------------------|
| 19              | FATALERROR        | 51     | Invalid structure number in DASDL-generated code |
|                 |                   | 61     | Fatal recovery error                             |
|                 |                   | 62     | Move failed                                      |
|                 |                   | 63     | Compare failed                                   |
|                 |                   | 64     | Move words failed                                |
|                 |                   | 70     | Wrong block type                                 |
|                 |                   | 71     | Control file corrupted                           |
|                 |                   | 72     | Data set corrupted                               |
|                 |                   | 73     | Set corrupted                                    |
|                 |                   | 80     | DBP terminated while in use                      |
|                 |                   | 81     | DBP terminated while in use                      |
|                 |                   | 82     | DBP terminated while in use                      |
|                 |                   | 83     | DBP terminated while in use                      |
|                 |                   | 85     | Reorg field invalid                              |
|                 |                   | 86     | EOF on Reorg work file                           |
|                 |                   | 87     | I/O error on Reorg work file                     |
|                 |                   | 90     | Invalid BCT parameter                            |
|                 |                   | 91     | Invalid lock on control file                     |
|                 |                   | 92     | Structure state invalid                          |
|                 |                   | 93     | In-use flag invalid                              |
|                 |                   | 94     | DBP function invalid                             |
|                 |                   | 95     | Control file not open                            |



**Table 8-1. Exception Categories (Cont.)**

| Category Number | Sub-Category Name | Number | Description                                                                                 |
|-----------------|-------------------|--------|---------------------------------------------------------------------------------------------|
| 19              | FATALERROR        | 96     | Database recovery required                                                                  |
|                 |                   | 97     | DBP tables corrupted                                                                        |
|                 |                   | 98     | DBP dynamic memory corrupted                                                                |
|                 |                   | 99     | Unknown error                                                                               |
| 20              | INTEGRITY ERROR   | 01     | Invalid control information                                                                 |
|                 |                   | 02     | Key data mismatch                                                                           |
|                 |                   | 03     | Set entry missing                                                                           |
|                 |                   | 04     | Set relative record number bad                                                              |
| 61              | RECOVERY          | 22     | Recovery open error                                                                         |
|                 |                   | 23     | Recovery close error                                                                        |
|                 |                   | 24*    | A read error occurred on the audit trail                                                    |
|                 |                   | 25*    | A write error occurred on block one of the audit file (halt/load recovery only)             |
|                 |                   | 26*    | A CHECKSUM error occurred on the audit trail                                                |
|                 |                   | 27     | Format level error                                                                          |
|                 |                   | 30     | Unable to find first half of split record or other control records which are known to exist |
|                 |                   | 31     | No halt/load recovery record is found from a previous recovery                              |
|                 |                   | 32     | Cannot find desired block in the file it should be in                                       |
|                 |                   | 33     | Missing discontinuity record                                                                |



Table 8-1. Exception Categories (Cont.)

| Category Number | Sub-Category Name | Number | Description                                                                        |
|-----------------|-------------------|--------|------------------------------------------------------------------------------------|
| 61              | RECOVERY          | 34     | Invalid lock field on old control file                                             |
|                 |                   | 35     | Invalid INUSE field on old control file                                            |
|                 |                   | 36     | Mismatch between old and new control files                                         |
|                 |                   | 37     | Discontinuity in database                                                          |
|                 |                   | 38     | Discontinuity in structure                                                         |
|                 |                   | 40     | Begin control word starts at offsets 1 through 7, or does not exist                |
|                 |                   | 41*    | Block one has a bad name, or there was an error in the previous audit file         |
|                 |                   | 50     | Link to second controlpoint points more than two controlpoints back                |
|                 |                   | 51     | Link does not point back to a controlpoint record                                  |
|                 |                   | 60     | Record split flags do not match                                                    |
|                 |                   | 61     | Begin and end control words of audit record are not identical                      |
|                 |                   | 62     | Audit block contains wrong audit block number                                      |
|                 |                   | 63     | Mismatch of date-timestamps across blocks                                          |
|                 |                   | 64*    | Mismatch of date-timestamps across files                                           |
|                 |                   | 65     | In applying after images, the before and structure file block images are not equal |





**Table 8-1. Exception Categories (Cont.)**

| <b>Category Number</b> | <b>Sub-Category Name</b> | <b>Number</b> | <b>Description</b>                                                                          |
|------------------------|--------------------------|---------------|---------------------------------------------------------------------------------------------|
| 61                     | RECOVERY                 | 66            | In applying before images, the after and structure file block images are not equal          |
|                        |                          | 67            | A record has been skipped in the audit trail                                                |
|                        |                          | 68            | Audit file purge record mismatch                                                            |
|                        |                          | 69            | Recovery can't stop during period when PURGE was in progress                                |
|                        |                          | 70            | Write error on one of the database structures                                               |
|                        |                          | 71            | Read error on one of the database structures                                                |
|                        |                          | 72            | OPEN error on structure                                                                     |
|                        |                          | 73            | CLOSE error on structure                                                                    |
|                        |                          | 74            | A structure is locked out, recovery cannot open it                                          |
|                        |                          | 75*           | Recovery buffer handling failed                                                             |
|                        |                          | 76            | Structure record does not exist in old control file                                         |
|                        |                          | 77            | Wrong file for recovery                                                                     |
|                        |                          | 78            | Wrong parameter file                                                                        |
|                        |                          | 79            | Corrupted control file                                                                      |
|                        |                          | 80            | A control record was found when an audit image was expected                                 |
|                        |                          | 81            | Structure type specified in the audit record does not correspond with the audit record type |



**Table 8-1. Exception Categories (Cont.)**

| <b>Category<br/>Number</b> | <b>Sub-Category<br/>Name</b> | <b>Number</b> | <b>Description</b>                                                                      |
|----------------------------|------------------------------|---------------|-----------------------------------------------------------------------------------------|
| 61                         | RECOVERY                     | 82            | Information contained in the audit record is invalid                                    |
|                            |                              | 83            | The data block presence flag state is invalid for current audit record application      |
|                            |                              | 84            | The data block count field is invalid for the current audit record application          |
|                            |                              | 85*           | Problem with swapping structure files for online reconstruct                            |
|                            |                              | 86            | Online reconstruct reached end of audit trail without finding End-of-Reconstruct record |
|                            |                              | 90            | Recovery cannot cross the audit trail discontinuity caused by a PURGE                   |

## NOTES

The recovery process does not return to the user program Category 61 (RECOVERY) errors encountered by the DBP during recovery of a database. The recovery process displays these errors on the ODT, and you can also find them in the DBP memory dump.

Category 61 (RECOVERY) errors marked with an asterisk require rebuild recovery to a prior point in the audit trail.



## SECTION 9

### DETAILS OF DATABASE FILES

#### INTRODUCTION

Most of the information in this section appears in tables that describe database file formats.

#### CONTROL FILE FORMAT

The database program (DBP) uses the control file to store information concerning run-time parameters of the DBP and the database structures.

The two types of records in the control file are:

- Header record
- Structure record

#### Header Record

The header record resides in Block one of the control file. Table 9-1 shows the format of the header record.

Table 9-1. Control File Header Record (Block One)

| Description                       | Digit Position | Digit Length |
|-----------------------------------|----------------|--------------|
| In-use flag                       | 00             | 01           |
| 0 = No recovery required          |                |              |
| 1 = Halt/load recovery required   |                |              |
| 2 = Rebuild required              |                |              |
| 3 = Lock-out failed               |                |              |
| 4 = Convert required              |                |              |
| 5 = Reorganization required       |                |              |
| 6 = Discontinuity required        |                |              |
| 7 = Purge required                |                |              |
| Number of records in control file | 01             | 04           |
| Version date-time stamp           | 05             | 15           |



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Details of Database Files

---

**Table 9-1. Control File Header Record (Block One)(Cont.)**

| Description                           | Digit Position | Digit Length |
|---------------------------------------|----------------|--------------|
| Creation date-time stamp              | 20             | 15           |
| Release level                         | 35             | 04           |
| Format level                          | 39             | 02           |
| Statistics                            | 41             | 01           |
| 0 = Reset                             |                |              |
| 1 = Set                               |                |              |
| Reserved                              | 42             | 01           |
| Reserved                              | 43             | 01           |
| Audit flag                            | 44             | 01           |
| 1 = DB not audited                    |                |              |
| 2 = DB audited (single audit trail)   |                |              |
| Number of TRANSACTIONS per syncpoint  | 45             | 04           |
| Number of syncpoints per controlpoint | 49             | 04           |
| Checksum of audit trail               | 53             | 01           |
| 0 = Reset                             |                |              |
| 1 = Set                               |                |              |
| Audit File ID                         | 54             | 12           |
| Next audit block number               | 66             | 12           |
| UPDATE-EOF for audit trail            | 78             | 06           |
| Filler                                | 84             | 26           |
| Control file lock status              | 110            | 02           |
| 0 = Not locked                        |                |              |
| 1 = Locked by DBP                     |                |              |
| 2 = Locked by DBP RECOVERY            |                |              |
| 3 = Locked by REORG                   |                |              |
| 4 = Locked by DMSTAT                  |                |              |
| 5 = Locked by CNV640                  |                |              |
| 6 = Locked by DMSDAP                  |                |              |
| 7 = Locked by DASDL                   |                |              |
| 8 = Locked by PURGE                   |                |              |
| 10 = Dummy lock                       |                |              |





**Table 9-1. Control File Header Record (Block One)(Cont.)**

| Description                                           | Digit Position | Digit Length |
|-------------------------------------------------------|----------------|--------------|
| Control file lock function                            | 112            | 02           |
| 0 = No lock function                                  |                |              |
| 1 = Halt/load recovery running                        |                |              |
| 2 = Rebuild running                                   |                |              |
| 3 = Reconstruct running                               |                |              |
| 4 = Rollback running                                  |                |              |
| 5 = Purge running                                     |                |              |
| 6 = Online dump running                               |                |              |
| 7 = Offline dump running                              |                |              |
| 10 = Dummy function                                   |                |              |
| Reserved                                              | 114            | 06           |
| Database program ID                                   | 120            | 12           |
| Last DMSTATUS result<br>(if FATALERROR occurred)      | 132            | 10           |
| ALLOWEDCORE value                                     | 142            | 08           |
| Recovery state                                        | 150            | 01           |
| 0 = No recovery in progress                           |                |              |
| 1 = Loading files                                     |                |              |
| 2 = Starting recovery                                 |                |              |
| 3 = Forcing sync                                      |                |              |
| 4 = Locking files                                     |                |              |
| 5 = Swapping files                                    |                |              |
| 6 = Halt/load recovery in progress                    |                |              |
| 7 = Writing audit record                              |                |              |
| 8 = Verifying files                                   |                |              |
| 9 = Applying audit trail images<br>to database        |                |              |
| Date-time stamp of online dump                        | 151            | 15           |
| Pointer to start online dump record<br>in audit trail | 166            | 30           |
| Audit File ID                                         | 166            | 12           |
| Audit Block Number (ABN)                              | 178            | 12           |
| Digit offset of record within block                   | 190            | 06           |
| Date-time stamp of last recovery                      | 196            | 15           |



**Table 9-1. Control File Header Record (Block One)(Cont.)**

| <b>Description</b>                                                                                                             | <b>Digit<br/>Position</b> | <b>Digit<br/>Length</b> |
|--------------------------------------------------------------------------------------------------------------------------------|---------------------------|-------------------------|
| Online reconstruct status<br>0 = No online reconstruct<br>4 = Online reconstruct in progress<br>8 = Online reconstruct aborted | 211                       | 01                      |
| Pointer to End of Online Reconstruct<br>record in audit trail                                                                  | 212                       | 30                      |
| Audit File ID                                                                                                                  | 212                       | 12                      |
| Audit Block Number (ABN)                                                                                                       | 224                       | 12                      |
| Digit offset of record within block                                                                                            | 236                       | 06                      |
| Count of structures locked out                                                                                                 | 242                       | 03                      |
| Reserved                                                                                                                       | 245                       | 55                      |
| Block control information (same format as<br>direct data set BCI with no presence<br>digits)                                   | 300                       | 60                      |

## Structure Records

The structure records begin at block 2. Block 2 is a structure record for Global Data that is not used. There is a structure record for every structure in the database; that is, a record for every data set, set, sub-set, and access described in the database as well as the statistics file. Table 9-2 shows the format of the structure record.



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Details of Database Files

---

**Table 9-2. Control File Structure Records (Blocks 2-n)**

| Description                              | Digit Position | Digit Length |
|------------------------------------------|----------------|--------------|
| Structure ID                             | 0              | 34           |
| Version date-time stamp                  | 34             | 15           |
| Discontinuity date-time stamp            | 49             | 15           |
| Last update date-time stamp 1            | 64             | 15           |
| Last update date-time stamp 2            | 79             | 15           |
| Structure status                         | 94             | 01           |
| 0 = Closed                               |                |              |
| 1 = Locked out                           |                |              |
| 2 = Opened                               |                |              |
| 3 = Empty                                |                |              |
| 4 = Opened but empty                     |                |              |
| 5 = Opened but no EOF                    |                |              |
| 6 = Deleted structure                    |                |              |
| Structure discontinuity required boolean | 95             | 01           |
| Number of system buffers                 | 96             | 02           |
| Number of user buffers                   | 98             | 02           |
| Number of READAHEAD buffers              | 100            | 02           |
| Structure number                         | 102            | 04           |
| Structure type                           | 106            | 02           |
| 1 = Control file structure               |                |              |
| 2 = Global data file structure           |                |              |
| 3 = Restart data set                     |                |              |
| 4 = Data set                             |                |              |
| 5 = Automatic set                        |                |              |
| 6 = Automatic subset                     |                |              |
| 7 = Manual subset                        |                |              |
| 8 = Access                               |                |              |
| 9 = Statistics file                      |                |              |
| 10 = Remaps Structure                    |                |              |
| 11 = Dummy                               |                |              |



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Details of Database Files

---

**Table 9-2. Control File Structure Records (Blocks 2-n) (Cont.)**

| Description                           | Digit Position | Digit Length |
|---------------------------------------|----------------|--------------|
| Structure organization                | 108            | 02           |
| 1 = Standard                          |                |              |
| 2 = Direct                            |                |              |
| 3 = Ordered                           |                |              |
| 4 = Embedded Unordered                |                |              |
| 5 = Random                            |                |              |
| 6 = Indexed sequential                |                |              |
| 7 = Index random                      |                |              |
| 8 = Ordered list                      |                |              |
| 9 = Unordered list                    |                |              |
| 10 = Bit vector                       |                |              |
| 11 = Direct access                    |                |              |
| 12 = Ordered access                   |                |              |
| 13 = Random access                    |                |              |
| 14 = Disjoint Unordered               |                |              |
| Embedded level of structure           | 110            | 02           |
| Record size in digits                 | 112            | 05           |
| Data size in digits                   | 117            | 05           |
| Control size in digits                | 122            | 05           |
| Number of AREAS                       | 127            | 03           |
| AREALENGTH in digits                  | 130            | 09           |
| BLOCKSIZE in digits                   | 139            | 05           |
| FAMILYNAME                            | 144            | 12           |
| Duplicates flag                       | 156            | 01           |
| 1 = Duplicates not allowed            |                |              |
| 2 = Duplicates allowed                |                |              |
| 3 = Duplicates first                  |                |              |
| 4 = Duplicates last                   |                |              |
| Structure number of parent structure  | 157            | 04           |
| Structure number of brother structure | 161            | 04           |
| Block offset to BCI                   | 165            | 05           |
| BLOCKSIZE in records                  | 170            | 04           |
| Structure CHECKSUM boolean            | 174            | 01           |
| Filler                                | 175            | 01           |
| Modulus (random data set)             | 176            | 08           |
| Family Attributes (see Table 9-3)     | 184            | 16           |
| Zero fillers in sets                  | 200            | 01           |
| WRITEAHEAD                            | 201            | 01           |
| Date-time stamp of last online dump   | 202            | 15           |





**Table 9-2. Control File Structure Records (Blocks 2-n)(Cont.)**

| <b>Description</b>                                                                                                                                                                                                                                                                                                                   | <b>Digit<br/>Position</b> | <b>Digit<br/>Length</b> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|-------------------------|
| State of online recovery<br>0 = No online reconstruct<br>4 = Online reconstruct in progress<br>8 = Online reconstruct aborted                                                                                                                                                                                                        | 217                       | 01                      |
| Recovery state of this structure<br>0 = No recovery<br>1 = To be recovered<br>2 = Structure file locked<br>3 = Structure file swapped<br>4 = Structure verified<br>5 = Force open<br>6 = Audit images being applied to<br>structure - normal<br>7 = Audit images being applied to<br>structure - online<br>8 = Structure dropped out | 218                       | 01                      |
| Data Set Referenced                                                                                                                                                                                                                                                                                                                  | 219                       | 04                      |
| Reserved                                                                                                                                                                                                                                                                                                                             | 223                       | 77                      |
| Block control information (same format as<br>direct data set BCI with no presence<br>digits)                                                                                                                                                                                                                                         | 300                       | 60                      |



The family attributes field describes where the file is to be stored, and by what assignment technique. This field can take one of three forms depending on the medium and assignment technique selected in DASDL. The different formats and meanings for this field appear in Table 9-3.

**Table 9-3. Control File Family Attributes Field**

| Field                | Length    | Values                                                                                          |
|----------------------|-----------|-------------------------------------------------------------------------------------------------|
| STRUCTURE ON PACK    |           |                                                                                                 |
| Multifile ID         | 12 digits | Diskpack family name                                                                            |
| Familykind           | 01 digit  | 1 = Diskpack                                                                                    |
| Pack Assignment Type | 01 digit  | 0 = By space<br>available<br>1 = By cylinder<br>boundary<br>2 = By area number<br>4 = By FIB-EU |
| Pack Single          | 01 digit  | 0 = Reset<br>1 = Set                                                                            |
| Filler               | 01 digit  | Unused                                                                                          |
| STRUCTURE ON DISK    |           |                                                                                                 |
| Multifile ID         | 12 digits | Blanks                                                                                          |
| Familykind           | 01 digit  | 2 = Disk                                                                                        |
| Disk Assignment Type | 01 digit  | 0 = By random number<br>1 = By file number<br>2 = By area number<br>4 = By FIB-EU               |



## STATISTICS FILE FORMAT

The operating system uses the statistics file for storage of statistics gathered by the DBP. This includes:

- Statistics on each database structure
- Statistics on each user program executed against the database

### Header Record

Block one of the statistics file is the same as other database files.

### Database Record

Block 2 of the statistics file contains statistics on the database as a whole. The format for this block appears in Table 9-4.

Table 9-4. Statistics File Database Record

| Description                              | Digit Position | Digit Length |
|------------------------------------------|----------------|--------------|
| EOF block                                | 0              | 08           |
| DBP BOJ date-time                        | 08             | 15           |
| DBP EOJ date-time                        | 23             | 15           |
| First user statistics record             | 38             | 06           |
| Next available user record               | 44             | 06           |
| Maximum available user statistics record | 50             | 06           |
| Total time DB was open in milliseconds   | 56             | 10           |
| Number DB opens                          | 66             | 06           |
| Number DB opens (update)                 | 72             | 04           |
| Number DB deadlocks                      | 76             | 06           |
| Number user terminates                   | 82             | 06           |



**Table 9-4. Statistics File Database Record (Cont.)**

| <b>Description</b>                                           | <b>Digit<br/>Position</b> | <b>Digit<br/>Length</b> |
|--------------------------------------------------------------|---------------------------|-------------------------|
| Number of Garbage Collections on DBP<br>dynamic memory       | 88                        | 08                      |
| Number of expand core requests                               | 96                        | 06                      |
| Number of return core requests                               | 102                       | 06                      |
| Maximum number of DB users at any one time                   | 108                       | 02                      |
| Current number of DB users                                   | 110                       | 02                      |
| Maximum size of dynamic memory in KD                         | 112                       | 04                      |
| Minimum size of dynamic memory in KD                         | 116                       | 04                      |
| Current size of dynamic memory in KD                         | 120                       | 04                      |
| Number of snapshots taken                                    | 124                       | 04                      |
| Number of users in DBP when snapshot<br>taken                | 128                       | 04                      |
| Number of users active in DBP when<br>snapshot taken         | 132                       | 04                      |
| Number of users waiting OPEN/CLOSE during<br>snapshot        | 136                       | 04                      |
| Number of users waiting locked records<br>during snapshot    | 140                       | 04                      |
| Number of users waiting for a block (I/O)<br>during snapshot | 144                       | 04                      |
| Number of users waiting locked block<br>during snapshot      | 148                       | 04                      |
| Number of users waiting on free list<br>during snapshot      | 152                       | 04                      |





**Table 9-4. Statistics File Database Record (Cont.)**

| <b>Description</b>                                                                        | <b>Digit<br/>Position</b> | <b>Digit<br/>Length</b> |
|-------------------------------------------------------------------------------------------|---------------------------|-------------------------|
| Number of users waiting on syncpoint during snapshot                                      | 156                       | 04                      |
| Amount of dynamic memory allocated during snapshot                                        | 160                       | 08                      |
| Amount of dynamic memory in use during snapshot                                           | 168                       | 08                      |
| Number of dynamic memory areas available during snapshot                                  | 176                       | 10                      |
| Reserved                                                                                  | 186                       | 114                     |
| Block control information (same format as direct data set on BCI with no presence digits) | 300                       | 60                      |

## **Structure Records**

The statistics structure records begin at block 3. There is a statistics structure record for every structure in the database. This includes a record for every data set, set, subset, and access described in the database, as well as the control file, global data file, and the statistics file. Table 9-5 shows the format for the statistics structure record.



**Table 9-5. Statistics File Structure Records (Blocks 3-n)**

| Description                                                                                       | Digit Position | Digit Length |
|---------------------------------------------------------------------------------------------------|----------------|--------------|
| Reserved                                                                                          | 0              | 08           |
| Number of opens (UPDATE)                                                                          | 08             | 04           |
| Number of physical opens                                                                          | 12             | 06           |
| Number of logical opens                                                                           | 18             | 06           |
| Number of physical reads                                                                          | 24             | 10           |
| Number of logical reads                                                                           | 34             | 10           |
| Number of physical writes                                                                         | 44             | 10           |
| Number of logical writes                                                                          | 54             | 10           |
| Number of FINDS, if data set<br>or INSERTS, if set                                                | 64             | 08           |
| Number of MODIFY actions, if data set<br>or REMOVES, if set                                       | 72             | 08           |
| Number of DELETES, if data set<br>or random FINDS, if set                                         | 80             | 08           |
| Number of new STORES, if data set<br>or sequential FINDS, if set                                  | 88             | 08           |
| Number of update STORES, if data set<br>or FIND KEY OF actions, if set                            | 96             | 08           |
| Number of block splits, if set/subset                                                             | 104            | 06           |
| Maximum number of buffers allocated                                                               | 110            | 02           |
| Number of times requested block was present<br>in memory                                          | 112            | 10           |
| Number of times idle buffer was used for<br>another block (requires one I/O<br>for read)          | 122            | 10           |
| Number of times changed buffer was used for<br>another block (requires two I/Os -<br>write, read) | 132            | 08           |
| Number of times buffer was added                                                                  | 140            | 08           |
| Number of times a physical readahead<br>was fired                                                 | 148            | 10           |
| Number of times a logical readahead<br>was fired                                                  | 158            | 10           |
| Number of times a writeahead is done                                                              | 168            | 10           |
| Number of times a wait for readahead<br>complete was issued                                       | 178            | 10           |
| Number of times a wait for writeahead<br>complete was issued                                      | 188            | 10           |
| Filler                                                                                            | 198            | 102          |
| Block control information (same as direct<br>data set with no presence digits)                    | 300            | 60           |



## User Records

The DBP stores statistics about each user program that accesses the database in the user records. The user records follow the structure records and the DBP stores them until it reaches End Of File (EOF) (block number 2000). If the DBP reaches EOF, it resets the next user record pointer to the beginning of the user records. The DMSTAT program will also reset this pointer, if you request the user statistics function. For more information about DMSTAT, refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities*.

Table 9-6 shows the format of the user records.

**Table 9-6. Statistics File User Record Format**

| Description                                                          | Digit Position | Digit Length |
|----------------------------------------------------------------------|----------------|--------------|
| User Program ID                                                      | 0              | 12           |
| Date/time user program opened DB                                     | 12             | 15           |
| Date/time user program closed DB                                     | 27             | 15           |
| Number of times user waited on locked record                         | 42             | 04           |
| Number of times user waited on block (I/O)                           | 46             | 04           |
| Number of times user waited on locked block                          | 50             | 04           |
| Number of times user waited on free list                             | 54             | 04           |
| Number of times user waited on restart                               | 58             | 04           |
| Number of times user waited on syncpoint                             | 62             | 04           |
| Number of times user caused a physical open of a database structure  | 66             | 04           |
| Number of times user caused a logical open of a database structure   | 70             | 04           |
| Number of times user caused a physical close of a database structure | 74             | 04           |
| Number of times user caused a logical close of a database structure  | 78             | 04           |
| Number of times user caused a physical read                          | 82             | 08           |
| Number of times user caused a logical read                           | 90             | 08           |
| Number of times user caused a physical write                         | 98             | 08           |
| Number of times user caused a logical write                          | 106            | 08           |
| Number of FINDS executed by user                                     | 114            | 08           |
| Number of MODIFY actions executed by user                            | 122            | 08           |
| Number of DELETES executed by user                                   | 130            | 08           |
| Number of new STORES executed by user                                | 138            | 08           |
| Number of update STORES executed by user                             | 146            | 08           |
| Number of automatic INSERTS caused by user                           | 154            | 08           |
| Number of manual INSERTS caused by user                              | 162            | 08           |



**Table 9-6. Statistics File User Record Format (Cont.)**

| Description                                                                    | Digit<br>Position | Digit<br>Length |
|--------------------------------------------------------------------------------|-------------------|-----------------|
| Number of automatic REMOVES caused by user                                     | 170               | 08              |
| Number of manual REMOVES caused by user                                        | 178               | 08              |
| Number of transactions caused by user                                          | 186               | 08              |
| Number of exceptions caused by user                                            | 194               | 08              |
| Number of times a physical READAHEAD<br>was issued                             | 202               | 08              |
| Number of times a logical READAHEAD<br>was issued                              | 210               | 08              |
| Number of times a WRITEAHEAD is issued                                         | 218               | 08              |
| Number of times a wait for READAHEAD<br>complete was issued                    | 226               | 08              |
| Number of times a wait for WRITEAHEAD<br>complete was issued                   | 234               | 08              |
| Reserved                                                                       | 242               | 58              |
| Block control information (same as direct<br>data set with no presence digits) | 300               | 60              |

## MEMORY FILE FORMAT

The DASDL compiler creates a memory file, so that the DBP can manage memory effectively. This file contains the parameters and other tables needed by the DBP at run time. When you initiate the DBP, it reads the contents of the memory file and places the information in DBP memory.

At digit position 2 of this file, there is a date-time stamp (15 digits) giving the time that the DASDL Compiler created the memory file. The first five digits are the Julian date; the next 10 digits are the time in milliseconds past midnight. The DBP compares this date-time stamp to its own internal date-time stamp at Beginning Of Job (BOJ). If a version error occurs because of a date-time stamp mismatch and the correct memory file is not available, you can create a new memory file with a \$UPDATE DASDL compile.





## AUDIT TRAIL FILE FORMAT

The audit trail consists of a series of files the DBP uses to audit every change made to the database. The DBP names the audit file XXXMNN, where XXX represents the first three characters of the database name, M represents a letter A through M, for primary audit, and N through Z, for secondary (DUPLICATED) audit, and NN represents a 2-digit number 01 through 99. Thus, the first file the DBP uses is XXXA01 for primary and XXXN01 for secondary; then the DBP continues the audit trail in file XXXA02 (or XXXN02). In a like manner, XXXA99 (or XXXN99) will wrap around to XXXB01 (or XXXO01), and XXXM99 or (XXXZ99) will wrap around to XXXA01 (or XXXN01).

A file name change occurs on the audit trail under the following conditions:

- The current audit file is not on disk or the appropriate diskpack.
- The current audit file is full.
- A halt/load, rebuild, rollback, or reconstruct recovery occurs.
- An I/O error occurs on the audit file.

If a full audit file condition occurs, the DBP forces a double controlpoint on the next syncpoint. This procedure ensures that all buffers are flushed and that any future recovery will not require the previous audit file.

Once the DBP fills and closes these files, you may back them up to another medium and then remove them. Should it be necessary to recover the database, the recovery routine will ask for the needed files which can then be copied from the backup medium.

The control file determines the proper file to start on the audit trail. Thus, if a database is reloaded from backup, the audit trail will start back with another audit file. For example, assume that the database TESTDB was used in testing mode to create records in the database files and this caused two audit trail files to be generated, TESA01 and TESA02. If the initialized files generated by DASDL are then reloaded from a backup, the audit trail will start over at TESA01. If the previous version of this file is still on disk or diskpack, a DUPLICATE FILE message will appear. You must remove the previous audit file (with the REMOVE command) before the DBP can continue.



## Header Record

Block one of each audit file consists of various control fields. Table 9-7 shows the format for this block.

**Table 9-7. Audit File Header Record (Block One)**

| Description                                     | Digit Position | Digit Length |
|-------------------------------------------------|----------------|--------------|
| Database ID                                     | 0              | 12           |
| Audit File ID                                   | 12             | 12           |
| EOF of audit file                               | 24             | 08           |
| Beginning block number                          | 32             | 12           |
| Ending block number                             | 44             | 12           |
| Prior I/O error Boolean                         | 56             | 01           |
| Prior audit I/O error result                    | 57             | 03           |
| Format level                                    | 60             | 02           |
| CHECKSUM flag                                   | 62             | 01           |
| Filler                                          | 63             | 01           |
| Filler (dependent upon BLOCKSIZE of audit file) | 64             | n            |
| Audit block trailer record                      | 64+n           | 60           |



## Trailer Record

The DBP appends the trailer record that appears at the end of Block one to every physical block in the audit trail. Table 9-8 shows the format for the audit file trailer record.

**Table 9-8. Audit File Trailer Record Format**

| Description                                           | Digit Position | Digit Length |
|-------------------------------------------------------|----------------|--------------|
| Block number                                          | 0              | 12           |
| First record split boolean                            | 12             | 01           |
| 0 = First record is not continued from previous block |                |              |
| 1 = First record is continued from previous block     |                |              |
| Last record split boolean                             | 13             | 01           |
| 0 = Last record is not continued on next block        |                |              |
| 1 = Last record is continued on next block            |                |              |
| Block offset to first begin control word (BCW)        | 14             | 06           |
| Block offset to last begin control word (BCW)         | 20             | 06           |
| Block offset to next available offset in block        | 26             | 06           |
| Date-time stamp                                       | 32             | 15           |
| CHECKSUM value                                        | 47             | 12           |
| Filler                                                | 59             | 01           |



## Event Records

Each block of the audit trail contains variable length records. Each variable length record is preceded and followed by a control word. This control word consists of a 2-digit record type and a 6-digit record size. The record size includes both control word sizes. The format for each record type appears in Table 9-9. Note that some record types are fixed in size while others are variable in size; these records may also be continued across blocks. If a record is continued across a block, the proper Booleans are set in the trailer record (see Table 9-8).

You may use the DMSAUD program to list an audit trail. Refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities* for more information about the DMSAUD program.

**Table 9-9. Audit File Syncpoint Record**

| Description                         | Digit Position | Digit Length |
|-------------------------------------|----------------|--------------|
| Begin control word                  | 0              | 08           |
| Record type (=01)                   | 0              | 02           |
| Record size (=46)                   | 02             | 06           |
| Link to second to last controlpoint | 08             | 30           |
| Audit File ID                       | 08             | 12           |
| Audit Block Number (ABN)            | 20             | 12           |
| Digit offset of record within block | 32             | 06           |
| End control word                    | 38             | 08           |
| Record type (=01)                   | 38             | 02           |
| Record size (=46)                   | 40             | 06           |

**Table 9-10. Audit File Controlpoint Record**

| Description                         | Digit Position | Digit Length |
|-------------------------------------|----------------|--------------|
| Begin control word                  | 0              | 08           |
| Record type (=02)                   | 0              | 02           |
| record size (=61)                   | 02             | 06           |
| Link to last controlpoint           | 08             | 30           |
| Audit File ID                       | 08             | 12           |
| Audit Block Number                  | 20             | 12           |
| Digit offset of record within block | 32             | 06           |
| Date-time stamp                     | 38             | 15           |
| End control word                    | 53             | 08           |
| Record type (=02)                   | 53             | 02           |
| Record size (=61)                   | 55             | 06           |





**Table 9-11. Audit File Database OPEN Record**

| <b>Description</b> | <b>Digit<br/>Position</b> | <b>Digit<br/>Length</b> |
|--------------------|---------------------------|-------------------------|
| Begin control word | 0                         | 08                      |
| Record type (=03)  | 0                         | 02                      |
| Record size (=47)  | 02                        | 06                      |
| Opening Program ID | 08                        | 12                      |
| Invocation number  | 20                        | 02                      |
| Task count         | 22                        | 02                      |
| Date-time stamp    | 24                        | 15                      |
| End control word   | 39                        | 08                      |
| Record type (=03)  | 39                        | 02                      |
| Record size (=47)  | 41                        | 06                      |

**Table 9-12. Audit File Database CLOSE Record**

| <b>Description</b> | <b>Digit<br/>Position</b> | <b>Digit<br/>Length</b> |
|--------------------|---------------------------|-------------------------|
| Begin control word | 0                         | 08                      |
| Record type (=04)  | 0                         | 02                      |
| Record size (=47)  | 02                        | 06                      |
| Closing Program ID | 08                        | 12                      |
| Invocation number  | 20                        | 02                      |
| Task count         | 22                        | 02                      |
| Date-time stamp    | 24                        | 15                      |
| End control word   | 39                        | 08                      |
| Record type (=04)  | 39                        | 02                      |
| Record size (=47)  | 41                        | 06                      |



**Table 9-13. Audit File Database Program BOJ Record**

| Description        | Digit Position | Digit Length |
|--------------------|----------------|--------------|
| Begin control word | 0              | 08           |
| Record type (=05)  | 0              | 02           |
| Record size (=31)  | 02             | 06           |
| Date-time stamp    | 08             | 15           |
| End control word   | 23             | 08           |
| Record type (=05)  | 23             | 02           |
| Record size (=31)  | 25             | 06           |

**Table 9-14. Audit File Database Program EOJ Record**

| Description        | Digit Position | Digit Length |
|--------------------|----------------|--------------|
| Begin control word | 0              | 08           |
| Record type (=06)  | 0              | 02           |
| Record size (=31)  | 02             | 06           |
| Date-time stamp    | 08             | 15           |
| End control word   | 23             | 08           |
| Record type (=06)  | 23             | 02           |
| Record size (=31)  | 25             | 06           |

**Table 9-15. Audit File Change Record**

| Description        | Digit Position | Digit Length |
|--------------------|----------------|--------------|
| Begin control word | 0              | 08           |
| Record type (=07)  | 0              | 02           |
| Record size (=31)  | 02             | 06           |
| Date-time stamp    | 08             | 15           |
| End control word   | 23             | 08           |
| Record type (=07)  | 23             | 02           |
| Record size (=31)  | 25             | 06           |



**Table 9-16. Audit File Recovery Record**

| Description                                   | Digit Position | Digit Length |
|-----------------------------------------------|----------------|--------------|
| Begin control word                            | 0              | 08           |
| Record type (=08)                             | 0              | 02           |
| Record size (=63)                             | 02             | 06           |
| Recovery type                                 | 08             | 02           |
| 1 = Abort recovery                            |                |              |
| 2 = Halt/load recovery                        |                |              |
| 3 = Rebuild recovery                          |                |              |
| 4 = Reconstruct                               |                |              |
| 5 = Rollback                                  |                |              |
| 6 = Online reconstruct                        |                |              |
| Link to syncpoint record in audit trail       | 10             | 30           |
| from which all transactions were backed-out   |                |              |
| Audit File ID of syncpoint record             | 10             | 12           |
| Audit file block number of syncpoint record   | 22             | 12           |
| Audit file record offset for syncpoint record | 34             | 06           |
| Date-time stamp                               | 40             | 15           |
| End control word                              | 55             | 08           |
| Record type (=08)                             | 55             | 02           |
| Record size (=63)                             | 57             | 06           |

**Table 9-17. Audit File Discontinuity Record**

| Description                   | Digit Position | Digit Length |
|-------------------------------|----------------|--------------|
| Begin control word            | 0              | 08           |
| Record type (=09)             | 0              | 02           |
| Record size (=36)             | 02             | 06           |
| Structure number              | 08             | 04           |
| Discontinuity type            | 12             | 01           |
| 1 = Database discontinuity    |                |              |
| 2 = Structure discontinuity   |                |              |
| 3 = PURGE discontinuity       |                |              |
| Discontinuity date-time stamp | 13             | 15           |
| End control word              | 28             | 08           |
| Record type (=09)             | 28             | 02           |
| Record size (=36)             | 30             | 06           |



**Table 9-18. Audit File STORE (New) Record**

| Description                                                          | Digit Position | Digit Length |
|----------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                   | 0              | 08           |
| Record type (=10)                                                    | 0              | 02           |
| Record size (variable)                                               | 02             | 06           |
| Structure number                                                     | 08             | 04           |
| Invocation number                                                    | 12             | 02           |
| Previous audit change number of data set block                       | 14             | 18           |
| Block number of new record                                           | 32             | 08           |
| Slot number of new record                                            | 40             | 04           |
| After image of new data set record (where n is data set record size) | 44             | n            |
| End control word                                                     | 44 + n         | 08           |
| Record type (=10)                                                    | 44 + n         | 02           |
| Record size (variable)                                               | 46 + n         | 06           |

**Table 9-19. Audit File DELETE (Old) Record**

| Description                                                               | Digit Position | Digit Length |
|---------------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                        | 0              | 08           |
| Record type (=11)                                                         | 0              | 02           |
| Record size (variable)                                                    | 02             | 06           |
| Structure number                                                          | 08             | 04           |
| Invocation number                                                         | 12             | 02           |
| Previous audit change number of data set block                            | 14             | 18           |
| Block number of deleted record                                            | 32             | 08           |
| Slot number of deleted record                                             | 40             | 04           |
| Before image of deleted data set record (where n is data set record size) | 44             | n            |
| End control word                                                          | 44 + n         | 08           |
| Record type (=11)                                                         | 44 + n         | 02           |
| Record size (variable)                                                    | 46 + n         | 06           |





**Table 9-20. Audit File Data Set Update Record**

| Description                                                               | Digit Position | Digit Length |
|---------------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                        | 0              | 08           |
| Record type (=12)                                                         | 0              | 02           |
| Record size (variable)                                                    | 02             | 06           |
| Structure number                                                          | 08             | 04           |
| Invocation number                                                         | 12             | 02           |
| Previous audit change number of data set block                            | 14             | 18           |
| Block number of updated record                                            | 32             | 08           |
| Slot number of updated record                                             | 40             | 04           |
| Before image of updated data set record (where n is data set record size) | 44             | n            |
| After image of updated data set record (where n is data set record size)  | 44 + n         | n            |
| End control word                                                          | 44 + 2n        | 08           |
| Record type (=12)                                                         | 44 + 2n        | 02           |
| Record size (variable)                                                    | 46 + 2n        | 06           |

**Table 9-21. Audit File Before Image of Data Set BCI Record**

| Description                                                 | Digit Position | Digit Length |
|-------------------------------------------------------------|----------------|--------------|
| Begin control word                                          | 0              | 08           |
| record type (=14)                                           | 0              | 02           |
| record size (variable)                                      | 02             | 06           |
| Structure number                                            | 08             | 04           |
| Invocation number                                           | 12             | 02           |
| Previous audit change number of data set block              | 14             | 18           |
| Data set block number of changed BCI                        | 32             | 08           |
| Offset into block for data set BCI                          | 40             | 06           |
| Size of BCI                                                 | 46             | 06           |
| Before image of data set BCI (where n is data set BCI size) | 52             | n            |
| End control word                                            | 52 + n         | 08           |
| record type (=14)                                           | 52 + n         | 02           |
| record size (variable)                                      | 54 + n         | 06           |



**Table 9-22. Audit File After Image of Data Set BCI Record**

| Description                                                   | Digit Position | Digit Length |
|---------------------------------------------------------------|----------------|--------------|
| Begin control word                                            | 0              | 08           |
| Record type (=15)                                             | 0              | 02           |
| Record size (variable)                                        | 02             | 06           |
| Structure number                                              | 08             | 04           |
| Invocation number                                             | 12             | 02           |
| Previous audit change number of data set block                | 14             | 18           |
| Data set block number of changed BCI                          | 32             | 08           |
| Offset into block for data set BCI                            | 40             | 06           |
| Size of BCI                                                   | 46             | 06           |
| After image of data set BCI<br>(where n is data set BCI size) | 52             | n            |
| End control word                                              | 52+n           | 08           |
| Record type (=15)                                             | 52+n           | 02           |
| Record size (variable)                                        | 54+n           | 06           |

**Table 9-23. Audit File Random Data Set STORE (New) Record**

| Description                                                                | Digit Position | Digit Length |
|----------------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                         | 0              | 08           |
| Record type (=16)                                                          | 0              | 02           |
| Record size (variable)                                                     | 02             | 06           |
| Structure number                                                           | 08             | 04           |
| Invocation number                                                          | 12             | 02           |
| Previous audit change number of data set block                             | 14             | 18           |
| Data set block number of changed block                                     | 32             | 08           |
| Data set slot number in changed block                                      | 40             | 04           |
| Data set fold offset in changed block                                      | 44             | 06           |
| Data set fold record from changed block                                    | 50             | 09           |
| After image of data set record<br>(where n is random data set record size) | 59             | n            |
| End control word                                                           | 59+n           | 08           |
| Record type (=16)                                                          | 59+n           | 02           |
| Record size (variable)                                                     | 61+n           | 06           |



**Table 9-24. Audit File Random Data Set DELETE (Old) Record**

| Description                                                             | Digit Position | Digit Length |
|-------------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                      | 0              | 08           |
| Record type (=17)                                                       | 0              | 02           |
| Record size (variable)                                                  | 02             | 06           |
| Structure number                                                        | 08             | 04           |
| Invocation number                                                       | 12             | 02           |
| Previous audit change number of data set block                          | 14             | 18           |
| Data set block number of changed block                                  | 32             | 08           |
| Data set slot number in changed block                                   | 40             | 04           |
| Data set fold offset in changed block                                   | 44             | 06           |
| Data set fold record from changed block                                 | 50             | 09           |
| After image of data set record (where n is random data set record size) | 59             | n            |
| End control word                                                        | 59+n           | 08           |
| Record type (=17)                                                       | 59+n           | 02           |
| Record size (variable)                                                  | 61+n           | 06           |

**Table 9-25. Audit File Record of Entry Added to Table**

| Description                                             | Digit Position | Digit Length |
|---------------------------------------------------------|----------------|--------------|
| Begin control word                                      | 0              | 08           |
| Record type (=20)                                       | 0              | 02           |
| Record size (variable)                                  | 02             | 06           |
| Structure number                                        | 08             | 04           |
| Invocation number                                       | 12             | 02           |
| Previous audit change number of set block               | 14             | 18           |
| Block number of new set entry                           | 32             | 08           |
| Slot number of new set entry                            | 40             | 04           |
| Existing entries moved right to make slot for new entry | 44             | 01           |
| 0= Existing entries moved left                          |                |              |
| 1= Existing entries moved right                         |                |              |
| Omega flag (set to 1 if entry deleted is omega key)     | 45             | 01           |
| Number of entries moved to accommodate new entry        | 46             | 04           |
| After image of new entry (where n is set record size)   | 50             | n            |
| End control word                                        | 50+n           | 08           |
| Record type (=20)                                       | 50+n           | 02           |
| Record size (variable)                                  | 52+n           | 06           |



**Table 9-26. Audit File Record Deleting Entry from Table**

| Description                                               | Digit Position | Digit Length |
|-----------------------------------------------------------|----------------|--------------|
| Begin control word                                        | 0              | 08           |
| Record type (=21)                                         | 0              | 02           |
| Record size (variable)                                    | 02             | 06           |
| Structure number                                          | 08             | 04           |
| Invocation number                                         | 12             | 02           |
| Previous audit change number of set block                 | 14             | 18           |
| Block number of deleted set entry                         | 32             | 08           |
| Slot number of deleted set entry                          | 40             | 04           |
| Direction in which existing entries<br>were moved         | 44             | 01           |
| 0= existing entries moved left                            |                |              |
| 1= existing entries moved right                           |                |              |
| Omega flag (set to 1 if entry deleted is<br>omega key)    | 45             | 01           |
| Number of entries moved to accommodate<br>new entry       | 46             | 04           |
| Before image of new entry<br>(where n is set record size) | 50             | n            |
| End control word                                          | 50+n           | 08           |
| record type (=21)                                         | 50+n           | 02           |
| record size (variable)                                    | 52+n           | 06           |

**Table 9-27. Audit File Record of Updating Table Entry**

| Description                                                       | Digit Position | Digit Length |
|-------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                | 0              | 08           |
| Record type (=22)                                                 | 0              | 02           |
| Record size (variable)                                            | 02             | 06           |
| Structure number                                                  | 08             | 04           |
| Invocation number                                                 | 12             | 02           |
| Previous audit change number of set block                         | 14             | 18           |
| Block number of updated entry                                     | 32             | 08           |
| Slot number of updated entry                                      | 40             | 04           |
| Before image of updated set entry<br>(where n is set record size) | 44             | n            |
| After image of updated set entry<br>(where n is set record size)  | 44+n           | n            |
| End control word                                                  | 44+2n          | 08           |
| Record type (=22)                                                 | 44+2n          | 02           |
| Record size (variable)                                            | 46+2n          | 06           |





**Table 9-28. Audit File Record for Table Split**

| Description                                                                    | Digit Position | Digit Length |
|--------------------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                             | 0              | 08           |
| Record type (=24)                                                              | 0              | 02           |
| Record size (variable)                                                         | 02             | 06           |
| Structure number                                                               | 08             | 04           |
| Invocation number                                                              | 12             | 02           |
| Previous audit change number of left table                                     | 14             | 18           |
| Previous audit change number of right table                                    | 32             | 18           |
| Block number of left table                                                     | 50             | 08           |
| Block number of right table                                                    | 58             | 08           |
| Slot number where table split occurred                                         | 66             | 04           |
| After image of left and right table<br>(where n is the table size for the set) | 71             | n            |
| End control word                                                               | 71+n           | 08           |
| Record type (=24)                                                              | 71+n           | 02           |
| Record size (variable)                                                         | 73+n           | 06           |

**Table 9-29. Audit File Record for Root Table Split**

| Description                                                           | Digit Position | Digit Length |
|-----------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                    | 0              | 08           |
| Record type (=25)                                                     | 0              | 02           |
| Record size (variable)                                                | 02             | 06           |
| Structure number                                                      | 08             | 04           |
| Invocation number                                                     | 12             | 02           |
| Previous audit change number of root table                            | 14             | 18           |
| Previous audit change number of left table                            | 32             | 18           |
| Previous audit change number of right table                           | 50             | 18           |
| Block number of root table                                            | 68             | 08           |
| Block number of left table                                            | 76             | 08           |
| Block number of right table                                           | 84             | 08           |
| Slot number where table split occurred                                | 92             | 04           |
| After image of left table                                             | 96             | n            |
| After image of right table<br>(where n is the table size for the set) | 96+n           | n            |
| End control word                                                      | 96+2n          | 08           |
| Record type (=25)                                                     | 96+2n          | 02           |
| Record size (variable)                                                | 98+2n          | 06           |



**Table 9-30. Audit File Record of Before Image of Table BCI**

| Description                                          | Digit Position | Digit Length |
|------------------------------------------------------|----------------|--------------|
| Begin control word                                   | 0              | 08           |
| Record type (=26)                                    | 0              | 02           |
| Record size (variable)                               | 02             | 06           |
| Structure number                                     | 08             | 04           |
| Invocation number                                    | 12             | 02           |
| Previous audit change number of set block            | 14             | 18           |
| Set block number of changed BCI                      | 32             | 08           |
| Offset into block for set BCI                        | 40             | 06           |
| Size of BCI                                          | 46             | 06           |
| Before image of set BCI<br>(where n is set BCI size) | 52             | n            |
| End control word                                     | 52+n           | 08           |
| Record type (=26)                                    | 52+n           | 02           |
| Record size (variable)                               | 54+n           | 06           |

**Table 9-31. Audit File Record of After Image of Table BCI**

| Description                                         | Digit Position | Digit Length |
|-----------------------------------------------------|----------------|--------------|
| Begin control word                                  | 0              | 08           |
| Record type (=27)                                   | 0              | 02           |
| Record size (variable)                              | 02             | 06           |
| Structure number                                    | 08             | 04           |
| Invocation number                                   | 12             | 02           |
| Previous audit change number of set block           | 14             | 18           |
| Set block number of changed BCI                     | 32             | 08           |
| Offset into block for set BCI                       | 40             | 06           |
| Size of BCI                                         | 46             | 06           |
| After image of set BCI<br>(where n is set BCI size) | 52             | n            |
| End control word                                    | 52+n           | 08           |
| Record type (=27)                                   | 52+n           | 02           |
| Record size (variable)                              | 54+n           | 06           |



**Table 9-32. Audit File Record of Block One Update**

| Description                           | Digit Position | Digit Length |
|---------------------------------------|----------------|--------------|
| Begin control word                    | 0              | 08           |
| Record type (=30)                     | 0              | 02           |
| Record size (=58)                     | 2              | 06           |
| Structure number                      | 8              | 04           |
| Update type                           | 12             | 02           |
| 1= EOF update                         |                |              |
| 2= Free list update                   |                |              |
| 3= Root pointer update                |                |              |
| 4= First fine pointer update          |                |              |
| 5= Last fine pointer update           |                |              |
| 6= Number of levels update            |                |              |
| Invocation number                     | 14             | 02           |
| Previous audit change number of block | 16             | 18           |
| Value before change                   | 34             | 08           |
| Value after change                    | 42             | 08           |
| End control word                      | 50             | 08           |
| Record type (=30)                     | 50             | 02           |
| Record size (=58)                     | 52             | 06           |

**Table 9-33. Audit File Record of General Before Image**

| Description                                                           | Digit Position | Digit Length |
|-----------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                    | 0              | 08           |
| Record type (=31)                                                     | 0              | 02           |
| Record size (variable)                                                | 02             | 06           |
| Structure number                                                      | 08             | 04           |
| Invocation number                                                     | 12             | 02           |
| Previous audit change number of block                                 | 14             | 18           |
| Block number of changed block                                         | 32             | 08           |
| Offset into changed block                                             | 40             | 06           |
| Size of changed record                                                | 46             | 06           |
| Before image of changed record<br>(where n is size of changed record) | 52             | n            |
| End control word                                                      | 52+n           | 08           |
| Record type (=31)                                                     | 52+n           | 02           |
| Record size (variable)                                                | 54+n           | 06           |



**Table 9-34. Audit File Record of General After Image**

| Description                                                          | Digit Position | Digit Length |
|----------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                   | 0              | 08           |
| Record type (=32)                                                    | 0              | 02           |
| Record size (variable)                                               | 02             | 06           |
| Structure number                                                     | 08             | 04           |
| Invocation number                                                    | 12             | 02           |
| Previous audit change number of block                                | 14             | 18           |
| Block number of changed block                                        | 32             | 08           |
| Offset into changed block                                            | 40             | 06           |
| Size of changed record                                               | 46             | 06           |
| After image of changed record<br>(where n is size of changed record) | 52             | n            |
| End control word                                                     | 52+n           | 08           |
| Record type (=32)                                                    | 52+n           | 02           |
| Record size (variable)                                               | 54+n           | 06           |

**Table 9-35. Audit File Record of Data Write Error**

| Description                                  | Digit Position | Digit Length |
|----------------------------------------------|----------------|--------------|
| Begin control word                           | 0              | 08           |
| Record type (=33)                            | 0              | 02           |
| Record size (variable)                       | 02             | 06           |
| Structure number                             | 08             | 04           |
| Area number                                  | 12             | 03           |
| Block number                                 | 15             | 08           |
| Block image<br>(where n is Block image size) | 23             | n            |
| End control word                             | 23+n           | 08           |
| Record type (=33)                            | 23+n           | 02           |
| Record size (variable)                       | 25+n           | 06           |





**Table 9-36. Audit File Record of Data Read Error**

| Description        | Digit Position | Digit Length |
|--------------------|----------------|--------------|
| Begin control word | 0              | 08           |
| Record type (=34)  | 0              | 02           |
| Record size (31)   | 02             | 06           |
| Structure number   | 08             | 04           |
| Area number        | 12             | 03           |
| Block number       | 15             | 08           |
| End control word   | 23             | 08           |
| Record type (=34)  | 23             | 02           |
| Record size (31)   | 25             | 06           |

**Table 9-37. Audit File Record of Initialize to Random Data Set Block**

| Description                                                                         | Digit Position | Digit Length |
|-------------------------------------------------------------------------------------|----------------|--------------|
| Begin control word                                                                  | 0              | 08           |
| Record type (=35)                                                                   | 0              | 02           |
| Record size (variable)                                                              | 02             | 06           |
| Structure number                                                                    | 08             | 04           |
| Invocation number                                                                   | 12             | 02           |
| Previous Audit Change Number                                                        | 14             | 18           |
| Data set block number of block initialized                                          | 32             | 08           |
| Data set block offset of block initialized                                          | 40             | 06           |
| Data set block size of block initialized                                            | 46             | 06           |
| After image of data set block initialized<br>(where n is size of block initialized) | 52             | n            |
| End control word                                                                    | 52+n           | 08           |
| Record type (=35)                                                                   | 52+n           | 02           |
| Record size (variable)                                                              | 54+n           | 06           |



**Table 9-38. Audit File Record of PURGE Structure List**

| Description            | Digit Position | Digit Length |
|------------------------|----------------|--------------|
| Begin control word     | 0              | 08           |
| Record type (=40)      | 0              | 02           |
| Record size (variable) | 02             | 06           |
| PURGE structure list   | 08             | n            |
| End control word       | 8+n            | 08           |
| Record type (=40)      | 8+n            | 02           |
| Record size (variable) | 10+n           | 06           |

n = 4 \* (number of structures purged)

**Table 9-39. Audit File Record of PURGE Done**

| Description                     | Digit Position | Digit Length |
|---------------------------------|----------------|--------------|
| Begin control word              | 0              | 08           |
| Record type (=41)               | 0              | 02           |
| Record size (=31)               | 02             | 06           |
| Date-time stamp of end of PURGE | 08             | 15           |
| End control word                | 23             | 08           |
| Record type (=41)               | 23             | 02           |
| Record size (=31)               | 25             | 06           |

**Table 9-40. Audit File Record of Start Online Dump**

| Description                      | Digit Position | Digit Length |
|----------------------------------|----------------|--------------|
| Begin control word               | 0              | 08           |
| Record type (=42)                | 0              | 02           |
| Record size (variable)           | 02             | 06           |
| Partial dump flag                | 08             | 01           |
| Date-time stamp                  | 09             | 15           |
| Name of dump tape (6 characters) | 24             | 12           |
| List of structures dumped        | 36             | n            |
| End control word                 | 36+n           | 08           |
| Record type (=42)                | 36+n           | 02           |
| Record size (variable)           | 38+n           | 06           |

n = 4 \* (number of structures dumped)



**Table 9-41. Audit File Record of End Online Dump**

| Description                                                  | Digit Position | Digit Length |
|--------------------------------------------------------------|----------------|--------------|
| Begin control word                                           | 0              | 08           |
| Record type (=43)                                            | 0              | 02           |
| Record size (variable)                                       | 02             | 06           |
| Dump result                                                  | 08             | 01           |
| 1= Dump OK                                                   |                |              |
| 2= Dump partly OK                                            |                |              |
| 3= Dump failed                                               |                |              |
| Date-time stamp of end of dump                               | 09             | 15           |
| Audit address of corresponding<br>start online dump record   | 24             | 30           |
| Audit file ID                                                | 24             | 12           |
| Audit block number (ABN)                                     | 36             | 12           |
| Digit offset of record within block                          | 48             | 06           |
| Filler                                                       | 54             | 01           |
| Date-time stamp of corresponding<br>start online dump record | 55             | 15           |
| Name of dump tape (6 characters)                             | 70             | 12           |
| List of structures dumped                                    | 82             | n            |
| End control word                                             | 82+n           | 08           |
| Record type (=43)                                            | 82+n           | 02           |
| Record size (variable)                                       | 84+n           | 06           |

$n = 4 * (\text{number of structures dumped})$



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Details of Database Files

---

**Table 9-42. Audit File Record of BEGIN-TRANSACTION**

| Description                          | Digit<br>Position | Digit<br>Length |
|--------------------------------------|-------------------|-----------------|
| Begin control word                   | 0                 | 08              |
| Record type (=80)                    | 0                 | 02              |
| Record size (=20)                    | 02                | 06              |
| Invocation number                    | 08                | 02              |
| Number of users in transaction state | 10                | 02              |
| End control word                     | 12                | 08              |
| Record type (=80)                    | 12                | 02              |
| Record size (=20)                    | 14                | 06              |

**Table 9-43. Audit File Record of END-TRANSACTION**

| Description                          | Digit<br>Position | Digit<br>Length |
|--------------------------------------|-------------------|-----------------|
| Begin control word                   | 0                 | 08              |
| Record type (=81)                    | 0                 | 02              |
| Record size (=20)                    | 02                | 06              |
| Invocation number                    | 08                | 02              |
| Number of users in transaction state | 10                | 02              |
| End control word                     | 12                | 08              |
| Record type (=81)                    | 12                | 02              |
| Record size (=20)                    | 14                | 06              |





## SECTION 10

# DATABASE PROGRAM (DBP) INTERNAL STRUCTURES

### INTRODUCTION

The database program (DBP) is the program that performs all of the functions a user program needs to access and update a particular database. All user programs for a particular database will interface to the DBP, and the DBP will interleave the processing for the user programs as much as possible.

The base size of an audited DBP, including the initial 20 KD stack is approximately:

300 KD for the BIND MEDIUM program  
450 KD for the BIND LARGE program  
600 KD for the BIND FLAT program

The difference in DBP sizes is due to the varying degree of segmentation. The MEDIUM version does a moderate amount of overlaying, while the LARGE version overlays only seldom-used routines. The FLAT version does not overlay. These sizes are approximate and vary from release to release as new features are incorporated into the DBP.

### DBP MEMORY REQUIREMENT

As the number of structures increases, so does the memory requirement of the DBP. The increase in memory requirement occurs because the DBP must hold in memory certain information on each database structure. This information includes the following items:

#### STRUCTURE REFERENCE LIST

Each entry in this list consists of 12 digits; there is one entry for each set and subset that refers to a data set. These entries form a list for every structure that is referenced by another structure. For example, a data set with two sets will have two entries in the list: one entry for each set. A second data set would have another structure reference list with entries corresponding to its sets and subsets.

#### KEY STRUCTURE LIST

Each entry in the key structure list consists of 25 digits. There is one entry for each key and/or key data item field defined for each set, subset, or access. There is one key structure list for each set, subset, and access.

#### DATA INFORMATION BLOCK

The Data Information Block (DIB) consists of 408 digits. There is one DIB for each structure in the database except for accesses. The DIB consists of a File Information Block (FIB), a buffer status block, an 8-digit Relative Record Number (RRN), and the file label area.

#### EMBEDDED STRUCTURE LIST

Each entry consists of 24 digits. There is one entry for each data set, set, or subset embedded in another structure. A list of these entries exists for every structure that has embedded structures within it.



## STRUCTURE TABLE

A structure table consists of 569 digits. There is one structure table for every structure in the database. This includes accesses, the control file, and the global data file.

## TASK PARAMETER BLOCK

Each task parameter block consists of 678 digits. There is one task parameter block for each active user program. An active user program is a user program that has opened the database.

## CURRENT PATH

A current path consists of 52 digits. There is one current path for each user program that invokes a set or subset. Note that these invocations are not necessarily explicit invocations.

## CURRENT RECORD

Each current record consists of 52 digits. There is one current record for each user program that invokes a data set.

## USER LIST

Each user list consists of 200 digits. There is one user list for each structure that a user program invokes.

## BUFFERS

The DBP allocates the buffers necessary for each structure as they are needed. The size of the buffer is the BLOCKSIZE of its structure. The DBP allocates buffers as needed according to the BUFFERS you specified in the DASDL source. If you do not specify a BUFFERS statement for a structure, the DBP uses a default value of one per data set opened and three per set opened. Although the DBP may allocate additional buffers for each user program according to the BUFFERS statement, the user programs may share all buffers for a particular structure.

## BUFFER CONTROL STRUCTURE

Each buffer has associated with it a buffer control structure that is 88 digits in length.

## MISCELLANEOUS

The DBP stores additional parameters in the dynamic area that are variable in size. This includes user stacks, Structure Information Blocks (SIBs), and set selection expression strings. The DBP allocates user stacks to handle the multi-threading of user program requests. SIBs are in the dynamic area during a user program OPEN function. Set selection expression strings are in the dynamic area during a user program function that requires a set selection expression string (for more information see the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*).

## DYNAMIC MEMORY DESCRIPTOR

Associated with each portion of memory in use in the dynamic area is a 16-digit descriptor the DBP uses for memory management. There is also a descriptor for each portion that is not in use.

Table 10-1 shows the format of the preceding information. The DASDL compiler allocates the fields marked with an asterisk (\*) and includes them in the compile size of the DBP.



**Table 10-1. Database Program Structures**

| Field                     | Size<br>(digits) | Number of Fields                                                                                                                                            |
|---------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GLOBAL AUDIT PARAMETERS*  | 246              | 1 per DBP                                                                                                                                                   |
| AUDIT STRUCTURE*          | 48               | 1 per DBP if audited (+47<br>if duplicate audit)                                                                                                            |
| STRUCTURE REFERENCE LIST* | n * 12           | 1 per structure referenced<br>by another structure<br>(n=number of structures<br>referring to structure)                                                    |
| KEY STRUCTURE LIST*       | n * 26           | 1 per accessing structure<br>(n=number key item fields<br>and key data item<br>fields)                                                                      |
| DATA INFORMATION BLOCK*   | 408              | 1 per structure<br>(except accesses)                                                                                                                        |
| EMBEDDED STRUCTURE LIST*  | n * 24           | 1 per structure<br>containing<br>other structures                                                                                                           |
| STRUCTURE TABLE*          | 569              | 1 per structure                                                                                                                                             |
| TASK PARAMETER BLOCK      | 678              | 1 per active user                                                                                                                                           |
| CURRENT PATH              | n * 60           | 1 per active user<br>(n=number sets, subsets,<br>accesses invoked by user<br>program)                                                                       |
| CURRENT RECORD            | n * 60           | 1 per active user<br>(n=number data sets<br>invoked<br>by user program)                                                                                     |
| USER LIST                 | 200              | 1 per opened structure                                                                                                                                      |
| BUFFERS                   | n * b            | Up to 1 per active data set<br>Up to 3 per active set<br>3 buffers for audit trail<br>(n=number users)<br>(b=BLOCKSIZE of<br>structure)<br>(Default values) |
| BUFFER CONTROL STRUCTURE  | n * 84           | 1 per buffer<br>(n=number active buffers)                                                                                                                   |



## MEMORY SIZE ESTIMATE

You can make a close estimate with the above figures, as follows:

1. Multiply the number of structures by 6 KD. Be sure to include the control file, statistics file (if STATISTICS is SET), each data set, subset, and set, and the audit trail as a structure.
2. Multiply the number of database users by 1 KD. The number of users is the maximum number of user programs accessing the database that are expected in the mix at any one time.
3. For each structure there may be a number of system buffers and a number of user buffers (refer to BUFFERS statement in the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation*). For each opened structure, calculate as follows:
  - a. Multiply BLOCKSIZE by the number of system buffers.
  - b. Multiply BLOCKSIZE by the number of user buffers and the number of users.
4. Multiply the BLOCKSIZE of the audit trail by 3.
5. Add the figures obtained in steps 1 through 4 to the appropriate DBP base size.

The estimate you obtained in step 5 should give a maximum memory size for any one run of the DBP. In practice, the estimate should be less. However, there are two considerations that may cause the estimate to be low.

The first case occurs in step 3. In some cases, the DBP may allocate more buffers because it has locked out a buffer. This occurs only when two or more user programs request a function (such as STORE) at the same time; that is, the DBP is multiprocessing multiple functions. Also, if an indexed sequential table is split, the DBP needs up to four buffers to update the pointers. Use the STATISTICS option to monitor the maximum number of buffers. Refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 3: Utilities* for more details on the statistics available for analysis.

The second case where DBP memory exceeds the estimate is on an audited database. If the DBP is maintaining an audit trail, it writes changed buffers (either data set or set buffers) after it writes the audit block containing a record of those changes. This means that an extremely large audit block may contain many records of changes to a data set, thereby forcing many buffers to be in memory. Refer to the *B 2000/B 3000/B 4000/V Series DMSII Operations Reference Manual Volume 1: Database Creation* for more information on audit trail records.

The estimate you obtained for DBP memory size should help determine the STACKSIZE and ALLOWEDCORE values. These parameters specify dynamic memory size at Beginning Of Job (BOJ) and maximum dynamic memory size, respectively. ALLOWEDCORE restricts the size of the DBP. This restriction is handled by periodic status runs of the DBP. If, during a status run, the DBP determines that the dynamic memory size has exceeded ALLOWEDCORE, it closes buffers if possible, until the dynamic memory size is below ALLOWEDCORE.

You can set STACKSIZE and ALLOWEDCORE in the database declaration in the DASDL source. You can also use an SP command to set ALLOWEDCORE while the DBP is executing.





The MCP automatically schedules the appropriate DBP when a user program opens a database. The MCP attaches each additional user program that opens the same database to the DBP. Once all user programs have closed the database, the DBP writes the contents of all buffers to disk and terminates normally. It is important that the DBP terminate normally; any other termination will result in corrupted files. Do not enter DS to stop the DBP. If you must terminate a run, you should enter DS to terminate the user programs attached to the DBP (you can locate these programs by entering the SB command with the database name option). This process will enable the DBP to terminate normally.

## OTHER CONSIDERATIONS

A FLAT DBP saves input/output (I/O) wait time that would be spent waiting for overlays, and also saves processor time because the operating system spends less time determining whether a needed overlay is in memory. You should use a FLAT DBP when there is sufficient system memory. There is a 6000 KD limit on the running size of a DBP. Because this limit includes code plus dynamic memory, an increase in code size will result in a decrease in available dynamic memory.



## SECTION 11

### USER PROGRAM INTERFACE TO DMSII

#### INTRODUCTION

The user program interfaces with a database through a database Branch Communicate (BCT) instruction. The DMS2 Extension Module of the MCP handles the BCT and communicates to the appropriate database program (DBP) for a database.

The user program that interfaces with a Data Management System II (DMSII) database differs from existing programs in the following instances:

- There are no File Information Blocks (FIBs) for files belonging to the database.
- There is a record area for each occurrence of a data set or remap declared in the user program. Thus, if a database has four structures, all four of which are remapped one time, that database effectively has eight structures. Invoking the entire database and the remaps explicitly will result in eight record areas. Note that invoking the entire physical database will not cause remaps to be invoked; remaps can only be invoked through a logical database or by implicit invocation on 01 level in a COBOL program. These record areas are Mod 4 in length and start at a Mod 4 address. Maximum length is 40,000 digits and minimum length is 4 digits. The length will always be the length defined for a data set record.
- There is a Structure Information Block (SIB).
- There is a DMSTATUS register.
- There are methods to handle exception conditions (USE procedures or ON EXCEPTION clauses in COBOL).
- There is a user interface that enables the user program to request the DBP to perform DMSII functions.

The following paragraphs contain more detailed discussions of: SIBs, DMSTATUS register, exception conditions, and user interfaces.

#### STRUCTURE INFORMATION BLOCK (SIB)

The SIB is a data structure that consists of: a format level number (currently 01), a logical database identification record (if you defined a logical database in DASDL), and from 1 to 1599 structure identification records. The logical database and structure identification records all have a length of 25 digits.

Table 11-1 shows the format for the logical database identification record.



**Table 11-1. Logical Database Identification Record Format**

| Field Name                 | Length    | Values                                                                     |
|----------------------------|-----------|----------------------------------------------------------------------------|
| Record type                | 1 digit   | 1 = logical database identification record                                 |
| Logical database ID number | 3 digits  | 1...999 = the number of the logical database as assigned by DASDL compiler |
| Version date-time stamp    | 15 digits | The date-time stamp of this structure.                                     |
| Reserved                   | 6 digits  |                                                                            |

If a user program invokes a logical database, the DBP makes the logical database identification record the first entry in the SIB (after the format level).

Table 11-2 shows the format for each structure identification record.

**Table 11-2. Structure Identification Record Format**

| Field Name                           | Length    | Values                                                                                                                  |
|--------------------------------------|-----------|-------------------------------------------------------------------------------------------------------------------------|
| Record type                          | 1 digit   | 0 = structure identification record                                                                                     |
| Structure number                     | 4 digits  | 2...9999 = the structure as assigned by DASDL Compiler.                                                                 |
| Master remap ID                      | 3 digits  | The master remap ID number as assigned by DASDL Compiler.                                                               |
| Invocation number for this structure | 2 digits  | Number of occurrences of this structure number - master remap if combination is assigned by the host language compiler. |
| Version date-time stamp              | 15 digits | The structure's version date-time stamp.                                                                                |
| Reserved                             | 1 digit   |                                                                                                                         |



The structure number is the number the DASDL compiler assigns to a particular data set, set, subset, remap or access. The occurrence number shows the number of times a user program has invoked a particular structure. The master remap ID is the number assigned by the DASDL Compiler to a disjoint remap and the DBP uses it in all BCTs that refer to that remap or any remaps and associated sets or subsets embedded in that remap.

The DMS2 Extension Module passes the address of the SIB to the MCP at the time of a database OPEN BCT operation. The MCP then supplies the information in the SIB to the DBP when the DBP requests it. Using this information, the DBP can either physically or logically open the structure(s) the user program has invoked.

A logical open occurs when a user structure has already been opened by another user program. It consists of setting up information in the DBP, and not having the DBP perform an OPEN BCT operation.

The SIB is Mod 4 length and must start at a Mod 4 address. The maximum length is 39,980 digits (maximum of 1599 files) and the minimum length is 28 digits (2-digit format level, 25-digit structure information record, and 1 digit to make it Mod 4 length).

## DMSTATUS REGISTER

Each user program contains a special register called DMSTATUS that shows the result of each DMSII function call. The DMSTATUS register is 10 digits long. The DMS2 Extension Module sets DMSTATUS at the end of each data management statement.

DMSTATUS consists of three fields. The first, a 2-digit field, gives the category of the result. The second, a 2-digit field, gives the subcategory of the result. The third, a 6-digit field, gives the structure number of the structure to which the result applies. The structure number consists of a 4-digit database structure number and a 2-digit occurrence number. The occurrence number shows which invocation of a particular structure a user program requested. Section 8 provides more information about the categories and subcategories.

## EXCEPTION HANDLING METHODS

When an exception occurs, you may handle it as follows:

- You may declare an error procedure (DMERROR USE procedure in COBOL), which the DBP will call.
- You may use an ON EXCEPTION clause with the data management statement in COBOL.

If you specify neither a DMERROR USE procedure nor an ON EXCEPTION clause, the DBP will terminate the user program when an exception occurs.

If you use both a DMERROR USE procedure and an ON EXCEPTION clause, the ON EXCEPTION overrides the DMERROR USE procedure (the DBP does not execute the DMERROR USE procedure).

You may declare only one DMERROR USE procedure per user program.

The DBP calls the DMERROR USE procedure each time an exception occurs during execution of a data management function, unless an ON EXCEPTION clause is associated with the data management function. The DBP returns control from the DMERROR USE procedure to the statement following the data management function that encountered the exception.





The following restrictions apply to the DMERROR USE procedure:

- You cannot branch out of a DMERROR USE procedure. The only acceptable way to exit a DMERROR USE procedure is to execute the RETURN FROM DMERROR USE PROCEDURE BCT that the host language compiler generates as the last piece of code in the DMERROR USE procedure.
- While in a DMERROR USE procedure, you should not execute a statement that might cause the DBP to call the DMERROR USE procedure again. If this should occur, the DBP will terminate the user program.

## SET SELECTION EXPRESSION STRING

The set selection expression string is a string of digits that represent, in "preorder" notation, the expression the DBP is to evaluate to satisfy a user program request.

This string consists of two structures: code structure and value pool.

### Code Structure

The code structure will appear first in the set selection expression string and will consist of a variable number of "containers." There are two types of containers: logical and relational.

#### Logical Container

The DBP uses the logical container to reflect any AND, OR, NOT, or End Of String (EOS) conditions that are present in the set selection expression.

A logical container consists of a 1-digit field that defines the type of operation the DBP is to perform. The possible values for this digit are:

|   |   |                     |
|---|---|---------------------|
| 0 | = | EOS (End of String) |
| 7 | = | AND                 |
| 8 | = | OR                  |
| 9 | = | NOT                 |

#### Relational Container

Use the relational container to specify the relational operators, values, and value attributes of each key value expression. The key value expression is in the set selection expression of the host language source program.

The relational container is 11 digits long and consists of these five fields: relational OP, key item identifiers, value offset, value type, and value length. The following paragraphs describe each field.



### **Relational OP [1 digit]**

The relational OP field specifies the condition the DBP is to test. Possible values are:

|   |   |                                |
|---|---|--------------------------------|
| 1 | = | LSS (less)                     |
| 2 | = | EQL (equal)                    |
| 3 | = | LEQ (less than or equal to)    |
| 4 | = | GTR (greater)                  |
| 5 | = | NEQ (not equal)                |
| 6 | = | GEQ (greater than or equal to) |

### **Key Item Identifiers [2 digits]**

The key item identifier field consists of 2-digits that specify the number of the key item that is being referenced. The DASDL compiler assigns this number from the sequence of the key items in the database description.

The DBP uses the key item identifiers to link into a table of key item descriptions.

### **Value Offset [4 digits]**

The value offset field specifies the offset in the value pool where the DBP stores the value to be compared to the key item. The offset is relative to the start of the set selection expression string, not to the start of the value pool. If the code structure is 200 digits long, the first value in the value pool has an offset of 200.

### **Value Type [1 digit]**

The value type field specifies the data type of the value in the value pool. Possible values for this field are:

|   |   |    |
|---|---|----|
| 0 | = | UN |
| 1 | = | SN |
| 2 | = | UA |

The host language compiler ensures that value types are compatible between the key item and the value in the value pool.

The DBP puts the signs to the left for all signed values in the value pool.

You will get a syntax error if you try to compare an 8-bit field with a 4-bit field.

### **Value Length [3 digits]**

The value length field specifies the length of the value that the DBP compares to the key item. The DBP considers the length to be compatible with the type of the key item. If the type is UA, then the length will reflect characters; if UN, it will reflect digits; and if SN, it will reflect digits minus the sign.



## Value Pool Structure

The value pool structure immediately follows the code structure and contains all values referenced by the set selection expression string. The following examples illustrate the value pool structure.

Examples:

EXAMPLE 1:  $A > 6$  AND  $C > "AB"$  OR  $X = 10$

In Example 1, A is key item 1, C is key item 2, and X is key item 3. The value offsets for 6, AB, and 10 are 36, 38, and 42 respectively.

The set selection expression string would be:

8 7 40100360001 40200382002 20300420002 0

The value pool appears as follows:

60C1C210

EXAMPLE 2:  $A < 6$  OR  $C > -4$  OR ( $X = 10$  AND  $P = 4$ )

In Example 2, A is key item 1, C is key item 2, X is key item 3, and P is key item 4. The value offsets for 6, -4, 10, and 4 are 48, 49, 51, and 53, respectively.

The set selection expression string would be:

8 8 10100480001 40200491001 7 20300510002 20400530001 0

The value pool appears as follows:

6D410400



## INDEX

### A

ABORT, 8-7  
ABORT exception, 4-5  
ALLOWEDCORE, 10-4  
AUDIT  
    Audit Block Number, 4-3  
    Audit Change Numbers, 4-3  
    Audit Record Offset, 4-3  
    audit trail record, 4-3  
    Previous Audit Change Number (PACN), 4-3  
Audit Block Number (ABN), 4-8  
audit buffer, 4-3  
Audit File  
    After Image of  
        Data Set BCI Record, 9-24  
    Before Image of  
        Data Set BCI Record, 9-23  
    Change Record, 9-20  
    Controlpoint Record, 9-18  
    Data Set Update Record, 9-23  
    Database CLOSE Record, 9-19  
    Database OPEN Record, 9-19  
    Database Program BOJ Record, 9-20  
    Database Program EOJ Record, 9-20  
    DELETE (Old) Record, 9-22  
    Discontinuity Record, 9-21  
    Header Record (Block One), 9-16  
    Random Data Set  
        DELETE (Old) Record, 9-25  
        STORE (New) Record, 9-24  
    Record Deleting Entry From Table, 9-26  
    Record for Root Table Split, 9-27  
    Record for Table Split, 9-27  
Audit File  
    Record of  
        After Image of Table BCI, 9-28  
        Before Image of Table BCI, 9-28  
        BEGIN-TRANSACTION, 9-34  
        Block One Update, 9-29  
        Data Read Error, 9-31  
        Data Write Error, 9-30  
        End Online Dump, 9-33  
        END-TRANSACTION, 9-34  
        Entry Added to Table, 9-25  
        General After Image, 9-30  
        General Before Image, 9-29  
        Initialize  
            to Random Data Set Block, 9-31  
        PURGE Done, 9-32  
        PURGE Structure List, 9-32  
        Start Online Dump, 9-32  
        Updating Table Entry, 9-26  
    Recovery Record, 9-21  
    STORE (New) Record, 9-22  
    Syncpoint Record, 9-18  
    Trailer Record Format, 9-17  
audit file change record, 4-6

audit record, 4-7  
audit trail, 2-8, 9-15  
Audit trail files, 1-1, 1-3  
audit trail records, 10-4  
Audit Write Count (AWC), 4-3  
AUDITERROR, 8-6

### B

backup procedures, 5-1  
BCT  
    (see Branch Communicate)  
BEGIN-TRANSACTION, 4-9  
    statement with AUDIT, 4-9  
bind listing, 5-3  
BLOCKSIZE, 10-4, 3-13  
bound-in user program, 6-5  
Branch Communicate (BCT), 11-1  
Buffer Control Structure, 10-2  
Buffers, 10-2, 10-4, 2-20  
    flushing, 4-2

### C

CHECKSUM, 3-31  
CLOSE, 4-5  
CLOSEERROR, 8-6  
CLOSEERROR exception, 4-5  
code structure, 11-4  
Commands  
    IR, 2-2  
    SB, 2-2  
    SP, 2-2, 2-16  
Compiler Control Record  
    \$CODE, 6-2  
    \$DMSICM, 6-2  
    \$REORGANIZE, 3-11  
    REBIND PRODUCTION, 6-2  
Compression/Balancing, 3-11  
Control file, 1-1, 1-2, 3-8, 3-9, 9-15  
Control File  
    Family Attributes Field, 9-8  
    Header Record, 9-1  
    Structure Records (Blocks 2-n), 9-4  
controlpoint record, 4-2  
controlpoints, 4-2  
    frequency of, 4-2  
COPY BACK, 3-12  
COPY function, 5-1, 5-4  
COPY TO, 3-12  
CREATE, 4-9  
Current Path, 10-2  
Current Record, 10-2, 4-5





B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Index

---

## D

### DASDL

- File Naming Conventions, 1-1
- data buffer synchronization, 4-3
- Data Definition File, 1-1, 1-2
- Data Information Block (DIB), 10-1
- data item

- changes, 3-17

### Database

- Changes to features of, 3-31
  - Converting for new software, 3-29
  - dumping, 5-1
  - full dump, 5-6
  - full offline dump, 5-7
  - full online dump, 5-8
  - loading, 5-1, 5-6, 5-8
  - modifying, 3-1
  - recovering, 5-7, 5-9

- database activity (SB) command, 2-15

- database CLOSE, 4-9

- database CLOSE record, 4-2

### Database Maintenance

- Performing with DASDL (XXXREO), 3-11

- Performing with DMSREO, 3-12

### Database name, 2-3

- database program (DBP), 10-1, 1-1, 4-1, 6-1

### Database Program Structures, 10-3

- Database Program Tailored Code ICM, 1-2

### databases

- dumping and reloading
  - instructions, 5-6

- DATAERROR, 8-2

### DBP

- (see database program)

- DBP CLOSE, 4-6

- DBP OPEN, 4-6

- DDF, 3-8

- DEADLOCK, 8-1

- DELETE, 4-9

### DIB

- (see Data Information Block)

### DMERROR USE

- exiting instructions, 11-4

- Procedure, 11-3

- DMS2 Extension Module, 6-1

- DMS2 option, 2-1

- DMSTATUS register, 11-1, 11-3

- DS, 10-4

- DUMP function, 5-4

- DUMP option, 2-13, 5-3

### dumps

- types of, 5-3

- Duplicate audit trail files, 1-1

- DUPLICATED, 3-32

- DUPLICATED AUDIT, 3-32

- DUPLICATED audit trail, 1-3

- DUPLICATES, 8-1

- Dynamic Memory Descriptor, 10-2

## E

- embedded data sets, 3-23

- embedded sets and subsets, 3-23

- Embedded Structure List, 10-1

### END-TRANSACTION

- NO-AUDIT SYNC, 4-9

- statement with AUDIT, 4-9

- Error Messages, 7-1

### Example

- set selection expression string, 11-6

- value pool, 11-6

- Exception Categories, 8-1

- Exception Handling

- Methods, 11-3

## F

- FAMILYNAME, 3-31

- FATALERROR, 8-7, 8-10

### File Format

- Audit File, 9-15

- Control File, 9-1

- Database, 9-1

- Memory File, 9-14

- Statistics File, 9-9

### File Format Conversion

- examples, 3-13

- instructions, 3-13

- File Information Blocks (FIBs), 11-1, 6-5

- file name change, 9-15

- File Naming Conventions, 1-1

- FLAT DBP, 10-5

- FROM DATA SET, 3-12

- full database backup, 5-4

- full database dump, 5-3

- full dump, 5-3

## G

### Garbage Collection

- Compression, 3-11

- Set Balancing, 3-11

- GENERATE, 3-11

## H

- HALTLOAD option, 2-7

## I

- INTEGRITYERROR, 8-10

- INUSE, 8-6

- IOERROR, 8-4

- IR command, 2-3

- IR command options, 2-4

- DUMP, 2-13

- HALTLOAD, 2-7

- LOAD, 2-13



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Index

---

PRODUCTION, 2-12  
PURGE, 2-13  
REBUILD, 2-7  
RECONSTRUCT, 2-10  
REORGANIZE, 2-10  
RESTART, 2-14  
ROLLBACK, 2-8  
IR command parameters, 2-3  
IR DUMP command, 2-3, 5-3  
IR GENERATE command, 3-12

## K

key item identifier, 11-5  
Key Structure List, 10-1  
KEYCHANGED, 8-2

## L

LIMIT DMSDBP, 2-1  
LIMIT DMSTATUS record, 2-2  
LIMIT DMSUSERS record, 2-2  
LIMIT PRIORITYQ record, 2-2  
LIMIT records, 2-1  
LIMITERROR, 8-4  
LOAD option, 2-13, 5-1, 5-3, 5-4  
LOADFACTOR, 3-11, 3-13  
logical container, 11-4  
Logical Database  
    identification record, 11-1  
    Identification Record Format, 11-1

## M

master remap ID, 11-3  
MCP DMPOUT utility, 6-5  
Memory file, 1-1, 1-3, 3-8  
Memory Requirement  
    DBP, 10-1  
Memory Size  
    estimate, 10-4  
mix number, 2-15, 2-16  
MODIFY, 4-9

## N

NORECORD, 8-6  
NOTFOUND, 8-1  
NOTLOCKED, 8-2

## O

occurrence number, 11-3  
Offline Dump, 5-4  
OFFLINE DUMP option, 5-1  
ON EXCEPTION clause, 11-3  
Online Dump, 5-4  
ONLINE DUMP option, 5-1  
OPEN BCT operation, 11-3  
OPENERROR, 8-5

ORDERED BY, 3-12

## P

partial database backup, 5-5  
partial database dump, 5-3  
partial dump, 5-3  
partial offline dump  
    certain structures, 5-8  
partial online dump  
    certain structures, 5-8  
primary audit trail, 1-3  
production database program, 6-1  
    definition, 6-1  
Production DBP, 2-3, 6-1  
Production DBP  
    generating  
        instructions, 6-1  
        initiating, 6-4  
PRODUCTION option, 2-12, 6-5  
program abort recovery record, 4-5  
program switches, 6-4  
PURGE  
    recovery, 3-29  
    rules, 3-29  
PURGE option, 2-13  
PURGING, 3-29

## R

READONLY, 8-4  
REBUILD option, 2-7, 5-5  
Rebuild recovery, 5-5, 8-13  
rebuild recovery record, 4-7  
RECONSTRUCT option, 2-10, 5-5  
Record Format Conversion, 3-17  
    ADD/DELETE Items, 3-21  
    Data Item Changes, 3-21  
    Examples, 3-24  
    Instructions, 3-23  
    Item Options, 3-22  
    Record Format Change Pragmatics, 3-21  
    Structure Changes, 3-22  
    Version Checking, 3-20  
recovering  
    certain structures, 5-9  
RECOVERY, 8-10, 8-11, 8-13  
    Halt/Load, 4-5, 9-15  
    Halt/Load  
        Automatic, 4-6  
        Offline Reconstruct, 4-7  
        Online Reconstruct, 4-7  
        Program Abort, 4-4  
        Rebuild, 4-6, 4-7, 5-1, 9-15  
        Rebuild and Reconstruct, 4-6  
        Reconstruct, 4-7, 5-1, 9-15  
            Kinds, 4-7  
        Rollback, 4-8, 9-15  
        Types of, 4-4  
        using backup tapes  
            instructions, 5-4  
recovery record, 4-5, 4-7



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Index

---

REGULAR DBP, 3-8  
relational container, 11-4  
relational OP, 11-5  
Relative Record Numbers, 3-27  
Reorg DBP, 3-7, 3-8, 3-12  
REORG TAILORED CODE FILE (ICM), 3-8, 3-9  
Reorganization  
    capabilities, 3-7  
    Garbage Collection, 3-11  
    Limitations, 3-10  
    phases, 3-7  
    Precautions, 3-10  
    RESTART, 3-27  
    types of, 3-7  
Reorganization Information File, 3-12  
Reorganization Run, 3-8  
REORGANIZE option, 2-3, 2-10  
Required files, 5-1  
restart data set, 4-8  
restart data set records, 4-4  
restart data sets  
    use of  
        criteria, 4-8  
RESTART function, 3-27  
RESTART option, 2-14  
restart record, 4-5  
ROLLBACK FAILURE, 7-29  
ROLLBACK option, 2-8

## S

SB command, 2-15  
    examples, 2-15  
secondary audit trail  
    (see also DUPLICATED audit trail)  
set selection expression string, 11-4  
SIBs  
    (see Structure Information Blocks)  
SO DMS2 command, 2-1  
SP Change Options, 2-20  
SP command, 10-4, 2-16, 3-11  
SP Display Options, 2-16  
    ALLOWEDCORE, 2-19  
    AUDITFILE, 2-19  
    BUFFERS, 2-19  
    CONTROLPOINT, 2-19  
    LOADFACTOR, 2-19  
    STACKSIZE, 2-19  
    STATISTICS, 2-19  
    SYNCPOINT, 2-19  
SP Parameters, 2-16  
    CHANGE, 2-18  
    DISPLAY, 2-18  
    SET, 2-18  
    SWITCH, 2-18  
    SWITCH AUDITFILE, 2-19  
    STACKSIZE, 10-4  
Statistics file, 1-1, 1-2  
Database Record, 9-9  
Header Record, 9-9  
Structure Records (Blocks 3-n), 9-11  
User Record Format, 9-13

STATISTICS option, 10-4  
STORE, 10-4, 4-9  
Structure files, 1-1, 1-2  
Structure Identification  
    Record Format, 11-2  
    records, 11-1  
Structure Information Block (SIB), 11-1  
Structure Information Blocks (SIBs), 10-2  
structure information record, 11-3  
structure number, 11-3  
Structure Reference List, 10-1  
Structure Table, 10-1  
SUCCESSFUL, 8-1  
syncpoint, 4-1  
    CLOSE statement, 4-2  
    frequency of, 4-1  
syncpoint record, 4-2  
SYNTAX, 2-13, 3-27  
System Commands, 2-2  
    IR command, 2-2  
    SB command, 2-2  
    SP command, 2-2  
System Messages, 7-1  
SYSTEM/COPY TAPE format, 5-3  
SYSTEM/COPY utility, 5-1  
SYSTEM/COPY utility  
    backing up with, 5-5  
SYSTEMERROR, 8-2

## T

TAILORED CODE FILE, 3-8  
Tailored code ICM, 1-1  
Task Parameter Block, 10-2  
Transactions  
    BEGIN-TRANSACTION statement, 4-1  
    END-TRANSACTION statement, 4-1  
    partial, 4-1

## U

Update  
    \$UPDATE option, 3-2  
    add, 3-1  
    change audit trail attributes, 3-1  
    change options, 3-1  
    change parameters, 3-1  
    change physical attributes, 3-1  
    convert, 3-1  
    delete, 3-1  
    examples, 3-3  
    performing, 3-1  
UPDATE-EOF, 3-31  
USE DMS2 record, 2-1  
User List, 10-2  
user program ICM (DMSICM), 6-5  
USERLD, 4-6, 4-7

## V

value length, 11-5



B 2000/B 3000/B 4000/V Series DMSII  
Operations Reference Manual, Volume 2: Database Administration  
Index

---

value offset, 11-5  
value pool, 11-4  
value pool structure, 11-6  
value type, 11-5  
VERSIONERROR, 8-7

W

WRITEAHEAD, 2-20  
Written Audit Number (WAN), 4-3

